



**Optimal Control of an Uninhabited Loyal  
Wingman**

DISSERTATION

Clay J. Humphreys, Major, USAF

AFIT-ENY-DS-16-S-063

**DEPARTMENT OF THE AIR FORCE  
AIR UNIVERSITY**

***AIR FORCE INSTITUTE OF TECHNOLOGY***

**Wright-Patterson Air Force Base, Ohio**

Distribution Statement A. Approved for public release; distribution is unlimited

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This is an academic work and should not be used to imply or infer actual mission capability or limitations.

AFIT-ENY-DS-16-S-063

OPTIMAL CONTROL OF AN UNINHABITED LOYAL WINGMAN

DISSERTATION

Presented to the Faculty  
Graduate School of Engineering and Management  
Air Force Institute of Technology  
Air University  
Air Education and Training Command  
in Partial Fulfillment of the Requirements for the  
Degree of Doctor of Philosophy

Clay J. Humphreys, BS, MS  
Major, USAF

September 2016

Distribution Statement A. Approved for public release; distribution is unlimited

AFIT-ENY-DS-16-S-063

OPTIMAL CONTROL OF AN UNINHABITED LOYAL WINGMAN  
DISSERTATION

Clay J. Humphreys, BS, MS  
Major, USAF

Committee Membership:

Richard G. Cobb, PhD (Chairman)

David R. Jacques, PhD (Member)

Capt Jonah A. Reeger, PhD (Member)

## Abstract

As researchers strive to achieve autonomy in systems, many believe the goal is not that machines should attain full autonomy, but rather to obtain the right level of autonomy for an appropriate man-machine interaction. A common phrase for this interaction is manned-unmanned teaming (MUM-T), a subset of which, for unmanned aerial vehicles, is the concept of the loyal wingman. This work demonstrates the use of optimal control and stochastic estimation techniques as an autonomous near real-time dynamic route planner for the DoD concept of the loyal wingman. First, the optimal control problem is formulated for a static threat environment and a hybrid numerical method is demonstrated. The optimal control problem is transcribed to a nonlinear program using direct orthogonal collocation, and a heuristic particle swarm optimization algorithm is used to supply an initial guess to the gradient-based nonlinear programming solver. Next, a dynamic and measurement update model and Kalman filter estimating tool is used to solve the loyal wingman optimal control problem in the presence of moving, stochastic threats. Finally, an algorithm is written to determine *if* and *when* the loyal wingman should dynamically re-plan the trajectory based on a critical distance metric which uses speed and stochastics of the moving threat as well as relative distance and angle of approach of the loyal wingman to the threat. These techniques are demonstrated through simulation for computing the global outer-loop optimal path for a minimum time rendezvous with a manned lead while avoiding static as well as moving, non-deterministic threats, then updating the global outer-loop optimal path based on changes in the threat mission environment. Results demonstrate a methodology for rapidly computing an optimal solution to the loyal wingman optimal control problem.

AFIT-ENY-DS-16-S-063

*To the great 'I am'... 'Here am I'*

## Acknowledgements

The primary theme of this pursuit derives from a biblical passage, ‘...Let God transform you into a new person by changing the way you think...’ Romans 12:2.

In this small 3- year window, a special thank you to those with whom I worked while at AFIT. To my committee members, Dr. Jacques and Dr. Reeger, thank you for challenging me towards quality and applicable research. To fellow students and those who went before me, thank you for providing technical support and wisdom. A special thank you to my advisor, Dr. Richard Cobb for caring, pushing, and overall rooting for success with a quality product. Because of all of you, I do think differently.

Beyond the technical, I’d like to thank the many co-workers I have encountered during my career in the Air Force. My sincerest thank you to the numerous church mentors, pastors, teachers, and incredible friendships that I have encountered throughout my life that are a major part of shaping me. To my family: thank you, Mom, for letting me stretch my wings and fly, when I know every instinct told you not to. Thank you, Dad, for being an inspiration. Many of life experiences I view through a lens of how I will share it with you when we talk on the weekend. A word to my children: never feel like you have to achieve worldly pursuits such as these, but always know that you can. Finally, a thank you to my wife for your commitment and endurance and allowing me to feel that I possess all a man could ever desire.

There is a lifetime of love, support, and dedication from more people than I can possibly mention. To all of you, my sincerest thank you.

Clay J. Humphreys

# Table of Contents

	Page
Abstract .....	iv
Acknowledgements .....	vi
List of Figures .....	x
List of Tables .....	xii
List of Abbreviations .....	xiii
I. Introduction .....	1
1.1 Requirement .....	2
1.1.1 UAS Roadmap .....	2
1.1.2 2013-2038 USAF RPA Vector .....	3
1.1.3 Technology Horizons .....	3
1.1.4 Air Force Research Laboratory Autonomy Strategy .....	6
1.2 Loyal Wingman Definition and Candidate Scenario .....	6
1.2.1 Loyal Wingman Definition .....	6
1.2.2 Loyal Wingman Research Framework .....	7
1.2.3 Candidate Scenario .....	8
1.3 Research Objective and Contributions .....	9
1.4 Assumptions and Limitations .....	10
1.4.1 Mission Scenario Assumptions and Limitations .....	10
1.4.2 Loyal Wingman Model Assumptions and Limitations .....	11
1.4.3 Threat Assumptions and Limitations .....	12
1.5 Document Outline .....	12
II. Literature Review .....	14
2.1 MUM-T and Loyal Wingman Literature .....	14
2.2 UAV and Formation Control Techniques .....	16
2.2.1 Formation Rendezvous .....	17
2.2.2 Formation Hold .....	19
2.2.3 Formation Reconfiguration .....	22
2.2.4 Splinter Activities .....	23
2.2.5 Research Approach .....	27
2.3 Nonconvex Optimal Control .....	27
2.4 The Particle Swarm Optimization .....	29
2.4.1 Particle Swarm Optimization Background .....	29
2.4.2 PSO Parameters .....	31

	Page
2.4.3 PSO Seeds .....	31
2.4.4 PSO Constraints .....	32
2.4.5 PSO Model Order .....	32
2.4.6 PSO Conclusion .....	32
2.5 Models .....	32
2.5.1 Loyal Wingman .....	33
2.5.2 Threats .....	34
2.5.3 Multiple-Target Mission .....	38
2.6 Stochastic Estimation .....	40
2.6.1 Kalman Filter .....	40
2.6.2 Extended Kalman Filter .....	41
2.6.3 Unscented Kalman Filter .....	42
2.6.4 Particle Filter .....	42
2.7 Trajectory Re-Planning .....	43
2.8 Literature Review Conclusion .....	45
III. Formulate the Optimal Control Problem .....	46
3.1 The Optimal Control Problem .....	46
3.2 Components of the Loyal Wingman Problem	
Formulation .....	47
3.2.1 Dynamic Constraints .....	48
3.2.2 Boundary Conditions .....	48
3.2.3 Modified Inside-Outside Threat Product Function .....	50
3.3 Loyal Wingman Problem Formulation Summary .....	54
3.4 Initial Guess Results .....	56
3.5 Optimal Control Problem Formulation Conclusions .....	58
IV. Hybrid Optimization Methodology .....	60
4.1 The Loyal Wingman PSO Algorithm .....	60
4.1.1 PSO Seeds and Initialization .....	62
4.1.2 PSO Iterations .....	63
4.2 Results .....	66
4.3 Hybrid Methodology Conclusions and Recommendations .....	73
V. 3-D Results .....	75
5.1 Update Optimal Control Problem Formulation .....	75
5.2 3D Loyal Wingman Problem Formulation Summary .....	76
5.3 Update Loyal Wingman PSO Algorithm .....	78
5.3.1 The Reduced Order PSO Dynamic Model .....	78
5.3.2 PSO Seeds in Three Dimensions .....	79
5.4 Results .....	81
5.5 3-D Model Conclusions and Recommendations .....	88

	Page
VI. Dynamic, Non-Deterministic Threats . . . . .	89
6.1 Assumptions and Limitations . . . . .	89
6.2 Dynamic Threat Model . . . . .	89
6.3 Modeling Dynamic Threat Avoidance Region . . . . .	92
6.3.1 Computing Threat Exposure . . . . .	93
6.3.2 Measurement Update Model . . . . .	94
6.4 Results . . . . .	95
6.5 Non-Deterministic Threat Conclusions and Recommendations . . . . .	99
VII. Dynamic Re-Plan . . . . .	101
7.1 Mission Flow and Changes in Mission Environment . . . . .	101
7.2 Moving, Non-Deterministic Threats: Time to Re-Plan, $t_r$ . . . . .	104
7.3 Results . . . . .	107
7.4 Dynamic Re-Plan Conclusions . . . . .	109
VIII. Conclusions and Recommendations . . . . .	111
8.1 Research Questions . . . . .	111
8.2 Contributions . . . . .	113
8.3 Future Research Recommendations . . . . .	114
8.3.1 Improve the Particle Swarm Optimization Algorithm . . . . .	114
8.3.2 Extend the Moving, Stochastic Threat Model . . . . .	115
8.3.3 Improve Robustness . . . . .	115
8.4 Summary . . . . .	116
Appendix A. 2D Deterministic Method to Produce Seeds . . . . .	117
Appendix B. 3D Deterministic Method to Produce Seeds . . . . .	121
Appendix C. Experiment to Choose PSO Parameters . . . . .	124
Appendix D. Dynamic, Stochastic Threat Size and Shape . . . . .	126
Appendix E. Distance to Re-Plan Formulation . . . . .	128
Bibliography . . . . .	130
Vita . . . . .	143

## List of Figures

Figure	Page
2.1      3-D Threat Keep-Out Regions Using Superquadrics .....	39
3.1      Course Layout with Avoidable Threats .....	50
3.2      Course Layout with Unavoidable Threats .....	51
3.3      Modified Threat Exposure Function .....	52
3.4      Initial Guess Segments Supplied to the NLP .....	57
3.5      Nine Locally Optimal Solutions for Unavoidable Threats .....	58
4.1      PSO Algorithm Flowchart .....	61
4.2      Comparison of Fixed Time Through Unavoidable Threats .....	68
4.3      Comparison of Minimize Time Through Unavoidable Threats .....	69
4.4      Comparison of Fixed Time Through Avoidable Threats .....	70
4.5      Comparison of Minimize Time Through Avoidable Threats .....	70
4.6      Comparison of Minimize Time to Rendezvous Through Unavoidable Threats .....	71
4.7      Comparison of Minimize Time to Rendezvous Through Avoidable Threats .....	72
5.1      Plot Legend for Figures 5.2, 5.3, 5.4, and 5.5 .....	81
5.2      Fixed Time to Fixed Point .....	82
5.3      Minimize Time to Fixed Point .....	84
5.4      Minimum Time to Rendezvous with Lead Through Avoidable Threats .....	86
5.5      Minimum Time to Rendezvous with Lead Through Unavoidable Threats .....	87

Figure		Page
6.1	Coordinates and Distance Along a Known Road .....	90
6.2	Block Diagram of Threat Dynamics Model .....	91
6.3	Trajectory Simulation to Avoid Moving, Stochastic Threat with Fixed Point Rendezvous .....	96
6.4	Trajectory Simulation to Avoid Moving, Stochastic Threat with Lead Rendezvous .....	98
7.1	Dynamic Re-Planning Flowchart .....	102
7.2	Angle of Approach ( $\alpha$ ) to Threat Vector Normal .....	105
7.3	Demonstration of Re-Plan in Changing Mission Environment .....	108
A.1	Data Points Used to Fit Spline .....	118
A.2	49 Trajectories Representing 49 Control Vector ‘Particles’ .....	119
B.1	Points Generated in 3D for Spline Interpolation .....	122
D.1	Known and Unknown Threat Locations .....	127
E.1	Trajectory 1: Effect of $\rho$ , $\xi$ , and $H$ .....	129
E.2	Trajectory 2: Effect of $\rho$ , $\xi$ , and $H$ .....	129

## List of Tables

Table		Page
2.1	Possible Loyal Wingman Threats .....	38
3.1	Cost and Computation Time of 9 Initial Guesses .....	57
4.1	Cost and Computation Time Comparison for All Scenarios .....	73
5.1	Computation Time for Each Scenario .....	88
C.1	Average Cost of Various Constriction Factors .....	124
C.2	Average Cost of Various Social Weighting Factors .....	125

## List of Abbreviations

Abbreviation	Page
WMD	Weapons of Mass Destruction . . . . . 1
UAS	Unmanned Aerial Systems . . . . . 1
MUM-T	Manned-Unmanned Teaming . . . . . 1
TTP	Tactics, Techniques and Procedures . . . . . 1
USAF	United States Air Force . . . . . 3
RPA	Remotely Piloted Aircraft . . . . . 3
AFRL	Air Force Research Laboratory . . . . . 6
UAV	Unmanned Aerial Vehicle . . . . . 7
NLP	Nonlinear Programming . . . . . 14
PN	Proportional Navigation . . . . . 18
SMTG	Sliding Mode Terminal Guidance . . . . . 18
LQR	Linear Quadratic Regulator . . . . . 20
MPC	Model Predictive Controller . . . . . 21
MMPC	Multiplexed Model Predictive Control . . . . . 22
PSO	Particle Swarm Optimization . . . . . 22
GA	Genetic Algorithm . . . . . 23
NLP	Nonlinear Program . . . . . 24
RDTA	Robust Decentralized Task Assignment . . . . . 25
NLP	Nonlinear Program . . . . . 26
PSO	Particle Swarm Optimization . . . . . 29
DOF	Degree-of-Freedom . . . . . 33
MVEE	Minimum Volume Enclosing Ellipsoid . . . . . 35

Abbreviation		Page
SNR	Signal to Noise Ratio .....	36
SAA	Sense and Avoid .....	36
FAA	Federal Aviation Administration .....	36
NAS	National Air Space .....	36
EKF	Extended Kalman Filter .....	41
UKF	Unscented Kalman Filter .....	42
NOC	Neighboring Optimal Control .....	44
AFIT	Air Force Institute of Technology .....	143
C4ISR	Command, Control, Communication, Computers, Intelligence, Surveillance, Reconnaissance .....	143
NRO	National Reconnaissance Office .....	143

## OPTIMAL CONTROL OF AN UNINHABITED LOYAL WINGMAN

### I. Introduction

“... sensors detect abnormal movements of vehicles from a key Weapons of Mass Destruction (WMD) storage site [in an adversarial nation]. The U.N. authorizes interception of the WMD because proliferation and potential terrorist use of the WMD are greater risks than a likely response from [the adversary nation]. Penetrating, high-altitude airborne systems track the vehicle and provide cueing information to incoming strike aircraft. Launched from the off-shore aircraft carrier, the strike package comprises of manned tactical aircraft with numerous combat support Unmanned Aerial Systems (UAS) providing tactical intelligence communication relay, jamming support, and strike support. The joint strike fighter operates as a command ship and works in concert with its supporting unmanned systems as a seamless network of strike and jamming aircraft. The strike package penetrates [the adversarial nation’s] airspace and intercepts, strikes, and stops the convoy...” [1]

The vignette above is quoted from the FY2013 Unmanned Systems Integrated Roadmap [1] and illustrates a future combat environment in which a single manned aircraft operates as a command ship working in concert with unmanned systems to complete a mission that is vital to the security interests of the U.S. U.S. “deployed forces have seen how effective unmanned systems can be in combat operations...” which “has created expectations for expanding the roles for unmanned systems in future combat scenarios [2].” The earlier published FY2011 Unmanned Systems Integrated Roadmap [2] highlighted seven challenges facing all military service departments, one of which is Manned-Unmanned Teaming (MUM-T), stating “DoD must continue to implement technologies and evolve Tactics, Techniques and Procedures (TTP) that improve the teaming of unmanned systems with the manned force” [2]. MUM-T is a concept which describes manned and unmanned systems working to-

gether to achieve a common operational mission objective, and includes the concept of the ‘*loyal wingman*’. Much like manned aircraft of modern times that fly in formation under the tactical command of their lead, the loyal wingman is a UAV that flies under tactical command of a manned lead aircraft both while in and out of formation. A review of DoD documents in Section 1.1 reveals a loyal wingman may be involved in numerous mission applications, command and control architectures and potential scenarios. Therefore, this chapter establishes a definition as well as a command and control structure and candidate scenario for the loyal wingman that drives the research performed in this work. This chapter additionally establishes research questions that, when answered throughout this work, provide a contribution to the existing body of knowledge.

## **1.1 Requirement**

Multiple DoD requirements documents are examined to ensure the defined command and control architecture and candidate scenario are applicable to real-world user needs.

### **1.1.1 UAS Roadmap.**

The 2011 Roadmap [2] lists capabilities that may be met by a loyal wingman such as defeating ground explosives from standoff distances, assuring mobility to support multiple points of entry, enabling movement and maneuver for projecting offensive operations and protecting austere combat posts. When determining a research objective, these roadmaps provide a broad range of missions from which to establish a loyal wingman command and control architecture and candidate mission scenarios.

### **1.1.2 2013-2038 USAF RPA Vector.**

The 2013-2038 United States Air Force (USAF) Remotely Piloted Aircraft (RPA) Vector [3] distinguishes the loyal wingman concept from such topics as swarms of unmanned vehicles. It is envisioned that a loyal wingman would be used to accompany a manned lead aircraft to accomplish such missions as ISR, air interdiction, attacks against adversary integrated air defenses, offensive counter air or act as a weapons “mule” increasing the airborne weapons availability to the shooter. The USAF RPA Vector [3] envisions a loyal wingman aiding in manned missions acting fully autonomously by 2030.

### **1.1.3 Technology Horizons.**

The US Air Force Technology Horizons, published by the chief scientist of the US Air Force provides key science and technology focus areas for scientists and engineers to pursue that will provide technologically achievable capabilities to enable the Air Force to contribute to the US joint force effectiveness in 2030 and beyond [4]. This document highlights one of three grand challenges for the Air Force in the science and technology area as the need of creating fractionated, composable, survivable, autonomous systems. A loyal wingman should address the following four attributes.

#### **1.1.3.1 Fractionated.**

Modern systems are composed of subsystems that are physically integrated to form a full system, such as the current aircraft strike systems which integrate communication, ISR, electronic warfare and strike capability into a single platform. The integration of multiple subsystems into a single system may result in performance and cost impacts such as limits on range and high production and operating costs. The loss of any one subsystem may result in mission failure, decreasing the survivability

of the system as a whole. Moving from integrated to fractionated, “the system is, in effect, separated into fractional elements and physically dispersed [4].” If the fractionation is done properly, the communication between the fractionated subsystems remains small enough to avoid detection or to continue to operate in the presence of jamming. There are multiple ways a loyal wingman could be used to realize the fractionated system. One example is to establish a formation of UAVs whose capabilities are the subsystems of a modern aircraft strike system. The manned lead acts as the pilot controlling the various subsystems, and the subsystems themselves are “fractionated” into physically dispersed elements of a formation. In order to perform a strike, the various fractionated subsystems could fly to and orient themselves in a way that is best suited for the mission as a whole. The ISR subsystem loyal wingman flies to the area that needs to be surveyed, while the electronic warfare subsystem wingman flies to a slightly different location and orients itself in a way to jam the enemy’s awareness. Meanwhile, the ordnance-carrying wingman may fly to its assigned location and orient itself most appropriately for unloading the ordnance [4]. A candidate scenario should support the use of a single loyal wingman as a fractionated subsystem supporting a large system as well as requiring a low amount of communication to reduce risk of detection.

#### **1.1.3.2 Composable.**

The Technology Horizons [4] describes composable systems as the ability to quickly assemble the appropriate capabilities into a package that may meet a specific mission requirement. If one considers a fractionated system, then a mission that requires reconnaissance may require only one loyal wingman. However, a mission that requires first locating a target, then striking the target, may combine multiple fractionated subsystems into a single passage in a rapid fashion. The loyal wingman concept

supports composability because any number of loyal wingmen with varying capabilities may be packaged together in near real-time to accomplish any set of proposed missions. Depending on the mission, these loyal wingmen may work in close communication or may be able to handle their individual missions autonomously with little, to no communication with other loyal wingmen or the manned lead.

#### **1.1.3.3 Survivable.**

The term survivable could pertain to either a machine or human subsystem. The failure of a subsystem in a fully integrated system more likely results in full mission failure. If, however, the system was composed of fractionated subsystems and one of those subsystems fail, the mission may continue after the loss. Survivability may also refer to the survivability of the human operator. An exceptional justification for unmanned systems are they allow humans to operate or lead a mission from a safe location. The loyal wingman may be sent into an austere and dangerous environment, which may put the UAV at risk, but will increase the survivability of the human.

#### **1.1.3.4 Autonomous.**

Current systems already attain a certain level of autonomy, but the Technology Horizons [4] suggests that an increase in the level of autonomy will produce a wider range of Air Force functions in the future. Key attributes of autonomy include complex decision making, autonomous mission planning, and the ability to self-adapt as the environment changes. A loyal wingman may fulfill this need by autonomously planning a mission that meets the manned leads commanded mission objectives and dynamically re-plan the mission in the face of a changing mission environment with a minimal amount of communication with the manned lead or other loyal wingman subsystems.

#### **1.1.4 Air Force Research Laboratory Autonomy Strategy.**

Derived from the previously mentioned Technology Horizons [4], the Air Force Research Laboratory (AFRL) established their autonomy science and technology strategy with a vision of “Intelligent machines seamlessly integrated with humans - maximizing mission performance in complex environments” [5]. The third of four goals is to ensure continuation of the mission in environments with kinetic and non-kinetic threats. In order to meet this goal, systems must be developed which are able to protect themselves and adapt to the environment faster than an adversary. The loyal wingman will support this vision by adjusting to a changing threat environment and appropriately re-planning its mission.

### **1.2 Loyal Wingman Definition and Candidate Scenario**

The documents discussed in Section 1.1 [1, 2, 3, 4] allow a broad framework for the command and control architecture as well as the specific use of a loyal wingman. These broad definitions allow scientists and engineers the freedom to research various tools and techniques for realizing loyal wingman technology. In order to identify a specific research objective, this section establishes a specific definition for the term loyal wingman and a command and control framework. In addition, a candidate mission scenario is established that when solved, answers the research questions produced herein.

#### **1.2.1 Loyal Wingman Definition.**

Throughout this work the term *loyal wingman* refers to an uninhabited aerial vehicle that flies autonomously under the tactical command of a manned lead air vehicle and includes terms such as *unmanned loyal wingman*, *uninhabited loyal wingman* and *loyal wingman*. The uninhabited loyal wingman is not a remotely piloted aircraft. In-

stances that reference a vehicle controlled or piloted by a man will specify “manned” as part of the description.

### **1.2.2 Loyal Wingman Research Framework.**

Morales [6] provides an informal survey to address who is in control (man or machine) of an uninhabited vehicle and under what conditions. The consensus is that a manned lead pilot is already saturated with tasks and wishes to push as many decisions as possible to their wingmen. Regarding an uninhabited loyal wingman, this implies as much autonomy in the unmanned system as is practical. Therefore a command and control framework consistent with modern manned operations is defined in which a manned lead provides clear and concise direction with very little need of communication bandwidth. The uninhabited loyal wingman must perform the following tasks:

- Autonomously compute a mission plan based on communication from lead
- Dynamically re-plan mission in event of a changing threat environment or changing mission requirement

Other frameworks are possible areas of research, but are not pursued. These other frameworks include the Unmanned Aerial Vehicle (UAV) acting as a central planner for all other UAVs in the formation. Such a framework would consider a conglomerate of tasks such that a computer onboard a single UAV computes mission paths for all UAVs in formation and communicates these paths to other UAVs in the formation to accomplish all mission tasks. An additional framework may be a collaborative control environment similar to that of swarming behaviors in which all UAVs in formation are in constant communication with each other canvassing which vehicle is accomplishing which task and continually working together to determine the optimal solution to accomplish all mission tasks.

### 1.2.3 Candidate Scenario.

A formation of aerial vehicles, including a manned lead accompanied by four unmanned loyal wingmen enter an austere environment. The manned lead receives a multiple-objective tasking that includes dropping ordnance on multiple locations. The manned lead distributes a task to a single loyal wingmen, which includes dropping ordnance on a single location and meeting back in the formation. The mission and several fixed- and variable-time and variable-location endpoint scenarios must be accomplished.

The mission occurs in an austere environment in which threats exist, such as anti-aircraft artillery. Many threats will be non-deterministic and dynamic, meaning the threats are only approximately known and change with time. There also will exist scenarios in which accomplishing the mission objectives while fully avoiding these threats is infeasible due to a fortified target; therefore, the loyal wingman autonomous mission planner must be able to receive the tasked mission plan and compute a mission path that accomplishes the primary mission objectives while either fully avoiding or minimizing exposure to threats, and if necessary, minimizing time to rendezvous.

Upon execution of the computed mission plan, intelligence may identify additional static or dynamic “pop-up” threats that were not known when the original path was computed. The loyal wingman must autonomously determine whether a change in the mission environment warrants a change in the original path and, if necessary, update the original path and continue the mission. As a means of realizing a more autonomous system, communication will be limited to the manned lead providing the mission objective as well as communication on the changing threat environment. The loyal wingman will only communicate its ability to accomplish the mission and there will be no communication between the loyal wingmen on their individual mission plans.

### 1.3 Research Objective and Contributions

Chapter II provides a summary review of literature relevant to solving the loyal wingman problem. This includes a look at MUM-T technical works which focus on cognitive systems engineering, defining where the man's task should end and the computer's task begin, as well as proposed solution techniques for solving various definitions of the loyal wingman problem. There is a vast amount of research available on techniques for controlling UAVs, in general, and a significant amount of that research may be applied to various loyal wingman definitions. Chapter II additionally reviews various methods for modeling static threats as well as modeling and estimating threats that are moving and non-deterministic. As a result of the review, it is proposed that the loyal wingman problem *as defined herein* can be formulated and solved using optimal control and stochastic estimation techniques. The objective of this research is to contribute to solving the DoD concept of the loyal wingman with an assumed level of autonomy using optimal control and stochastic estimation techniques. This will be shown by answering the following three research questions:

1. How do you formulate the uninhabited loyal wingman optimal control problem in the presence of static, deterministic threats?
2. How do you formulate the uninhabited loyal wingman optimal control problem with moving, non-deterministic threats?
3. How and when do you dynamically re-plan the uninhabited loyal wingman optimal control problem in the presence of various deterministic and non-deterministic pop up threats that may arise during execution of a pre-computed path?

By answering these three questions, several contributions are made that both add to the existing body of knowledge as well as spawn additional areas of research. These contributions include:

1. a methodology for a rapid near optimal solution;
2. potential use in a near real-time environment;
3. a near optimal solution for comparison against other methods.

## 1.4 Assumptions and Limitations

Assumptions and limitations stated throughout the work are now summarized.

### 1.4.1 Mission Scenario Assumptions and Limitations.

Manned pilots view the main purpose of having a manned loyal wingman is to look after and protect one-another in austere environments through ‘mutual support’; however, the loyal wingman is unable to provide mutual support because there are no suitable sensors on board and the capability does not exist for a UAV to engage in autonomous aerial combat maneuvering against a manned or unmanned enemy. Generally, the loyal wingman must avoid detection by enemy aircraft because it is assumed the loyal wingman cannot survive an aerial combat situation.

Although a primary benefit of the use of UAVs is to reduce the risk of putting a manned pilot in danger, the uninhabited loyal wingman is not considered attritable and effort must be made to ensure the uninhabited loyal wingman’s safe return to the formation.

Communication between a manned lead and the loyal wingman will generally be established on pre-defined terms that a computer may interpret. The “dictionary” in this research will be limited to intermediate waypoint and rendezvous times and objectives. There is no communication between the individual unmanned vehicles in the formation.

Computational resources on board the UAV are limited and allocated to many

subsystems; therefore, resource allocation for route planning is limited to initial computation and in cases when the mission environment warrants a trajectory re-plan. There will not be a high-frequency re-plan of the trajectory as this will consume valuable resources.

The loyal wingman must be able to make online, near real-time calculations. There is not a pre-defined mission or a library of pre-defined maneuvers or paths as in Carter [7]. The loyal wingman computes the mission when the task is provided by the manned lead. Although computational efficiency will be leveraged where it can be found, the goal of this research is not to attain the computation time needed for real-time implementation and it is expected that there may be other computational tools and better programming techniques that can be used to achieve true near real-time computation.

#### **1.4.2 Loyal Wingman Model Assumptions and Limitations.**

Due to the similarities with Smith [8], a modified version of the five-state, 3 degree of freedom model is used in which roll rate and normal acceleration rate are the controls. Because the mission will be performed in a local environment, a flat, non-rotating Earth is assumed. High frequency updates on the position of non-deterministic threats are available from a sensor onboard the loyal wingman. Low frequency updates on threat and mission environment are provided via a friendly ISR asset operating in the vicinity of the mission.

The work herein seeks to obtain the optimal path for the loyal wingman to accomplish the established mission; this is often referred to as outer-loop control. Following computation of the outer-loop optimal path, it is then necessary to utilize an inner-loop controller in order to appropriately track that path. Additional information on techniques such as sliding mode control [9] used by Harl [10] and the linear quadratic

regulator used by Stankovic [11] and Venkataramanan [12, 13] are available, however it is assumed these or any other inner-loop tracking techniques are capable of tracking the computed optimal path and are not pursued as part of this research. Wind is not modeled because the assumed inner-loop controller will be built with appropriate models to account for disturbances due to wind.

### 1.4.3 Threat Assumptions and Limitations.

It is assumed that threats can be modeled analytically using superquadrics [14]. For simplicity, dynamic threats will travel along a straight road whose coordinates are known *a priori*. The threat will remain on the road, but the location along the road at any time is not known. A linear dynamic model is developed and estimates may be assumed to have a Gaussian distribution.

## 1.5 Document Outline

Chapter I motivated the problem and established a definition and candidate scenario in order to narrow the specific area of research. Chapter II provides a review of literature supporting how this research contributes to the existing body of knowledge as well as provides a set of techniques for enabling a solution. In Chapter III, the optimal control problem is formulated for various scenarios and initial results highlight a challenge in the optimal control community with gradient-based numerical methods getting ‘stuck’ in locally optimal solutions. Chapter IV discusses a hybrid gradient-heuristic technique to ameliorate the challenge identified in Chapter III and results for a 3-D model are presented in Chapter V to indicate a rapid technique to provide an accurate locally optimal solution. Chapter VI answers research question two by identifying a dynamic and measurement update model as well an estimation tool to account for and model moving, non-deterministic threats. The final research question

is answered in Chapter VII by developing a mission algorithm flow-chart that computes and simulates a trajectory, and provides a formulation for determining if and when a trajectory re-plan should occur. Finally, Chapter VIII provides a summary of relevant contributions and recommendations for future work. The Appendices provide additional detail and support for various technical discussions throughout the document.

## II. Literature Review

MUM-T and the loyal wingman are mission applications ripe for exploring various tools and techniques that may be used to realize their concepts. This chapter begins with a review of works in the open technical literature that attempt to solve various aspects of the MUM-T problem. In order for the research herein to establish a methodology that will contribute to the existing body of knowledge, a review is first performed to examine the control techniques that are used across the four attributes of UAV formation flying. Next, a discussion on nonconvex optimal control leads to a hybrid technique, using a particle swarm optimization algorithm to seed a Nonlinear Programming (NLP) solver. The remainder of the chapter provides a dynamic model for the loyal wingman, a review of methods for modeling threats, as well as a discussion on stochastic estimation techniques for dynamic trajectory re-planning.

### 2.1 MUM-T and Loyal Wingman Literature

The DoD established a roadmap for UAVs [1, 2] and established autonomy as a grand challenge [4]. Clough [15] highlights challenges to achieving autonomy in UAVs, while others such as Yang [16] provide a recommendation on the appropriate balance in autonomy to achieve the ‘right’ level of interaction between man and machine. Gangl [17] offers an experiment in the management of multiple UAVs from a single-seat fighter, while Schmitt [18] provides a mission planner able to support a pilot during MUM-T missions, and Jameson [19] recommends an approach for teams of vehicles to coordinate their activities without human oversight. Potential applications of such research was reviewed by Svenmarck [20]. Van Riper [21] and Durbin [22] each provide an assessment of operator workload on a current operational application of MUM-T: the Army Apache AH-64D. Additionally, the Navy successfully demon-

strated the first-ever carrier launch and recovery of an unmanned aircraft [23].

The U.S. Air Force is developing its version of MUM-T in the concept of the loyal wingman, technology that differs from swarming because the unmanned system will work with and interact with the manned aircraft in various tactical missions [4]. The loyal wingman serves a manned lead in two broad categories: mutual support and formation flying. Mutual support stems from the Airman’s Creed and the idea that one will “never leave an airman behind [24].” This gives rise to the idea of a loyal wingman supporting the manned lead through aerial combat. Carter [7] created a library of time optimal maneuvers for use by a remote pilot. McGrew [25] used approximate dynamic programming and Horie [7] used a two-sided optimization approach to solve the highly dynamic minimax problem existent in the pursuer-evader control problem. The computational complexity of the highly dynamic, non-deterministic mutual support mission makes it impractical to develop real-time, on-line optimal solutions and is therefore not pursued in the research herein.

The second broad category for supporting a manned lead is formation flying. Murphey [26] identifies a number of challenges associated with integrating unmanned systems into the joint force, to include formation flight for autonomous aerial refueling, which DARPA successfully demonstrated [27]. Waydo [28] specifically addressed the concept of the loyal wingman which describes a flight test using inner-loop optimal control techniques to demonstrate safe formation flight of a loyal wingman in the event of a loss of communication. The test objective was to maintain formation, but in the event of a loss of communication, the loyal wingman departed to a safe formation flying distance, then returned to normal formation distance once the communication was re-established. Waydo demonstrated optimal control techniques for maintaining formation, but did not indicate the loyal wingman’s ability to perform splinter activities beginning and ending in formation, nor does it consider avoidance

of threats.

Performed in support of the U.S. Army, Garcia's [29] research goal was to introduce a method to integrate UAS into a highly functional manned/unmanned team through design and implementation of 3-D distributed formation/flight control algorithms with the goal to act as wingmen for manned aircraft. Artificial potential functions simulated the ability of a group of unmanned loyal wingmen to initialize formation through a self-ordering algorithm, hold a standard formation, dynamically modify the formation and perform a splinter activity, which are the four attributes of formation flying. Additionally, non-deterministic threats were avoided using fuzzy reasoning. Additional discussion on the methods used by Garcia is found in Section 2.2.

The definitions and scenarios that are solved vary greatly in each of the works of this review, which indicates there is still a significant amount of research remaining for solving MUM-T applications. Garcia's [29] mission scenario most closely resembled the loyal wingman as defined herein. However, additional research is necessary to solve the loyal wingman problem as defined herein, because threats must be avoided and exposure minimized while meeting optimization criteria. Additionally, the use of fuzzy reasoning is not an optimal method for estimating the state of non-deterministic threats. The research herein provides a contribution to the engineering community by demonstrating an optimal control and stochastic estimation methodology for solving the loyal wingman problem *as defined herein*. The methodology will be chosen from a review of techniques used to control UAVs in the next section.

## 2.2 UAV and Formation Control Techniques

In order to address the research goals, this section reviews techniques that are used throughout literature to control UAVs and groups of UAVs. The desire is to determine a methodology that will aid in solving the loyal wingman candidate scenario

as defined herein that is unique in relation to other works that have addressed the loyal wingman. As a reminder, Section 1.4 established that inner-loop control will not be addressed in the work herein. The methods discussed in the following four subsections address outer-loop techniques to control UAVs and groups of UAVs throughout the four attributes of formation flight. For the purposes of the ensuing discussion, outer-loop control will be further parsed into two categories, one in which periodic tracking updates provide for sequential control. The other category entails calculating the vector of control inputs for a full outer-loop path trajectory.

### 2.2.1 Formation Rendezvous.

Formation rendezvous is where two or more vehicles join together from physically separate locations into a formation that is intended to be held for some finite period of time<sup>1</sup>. Techniques for controlling a vehicle to a successful rendezvous include graph theory, the Dijkstra algorithm, proportional navigation, line of sight, sliding mode terminal guidance, and artificial potential functions, all of which are of the category of outer-loop control defined herein that include sequential path planning control.

Giuletti [30] discusses graph theory and the Dijkstra algorithm to define an optimal formation configuration. The objective function is set to optimize the communication and data transfer amongst the vehicles in the formation. Graph theory is utilized by supposing the individual aircraft are nodes on a graph and the lines of communication between them as the edges that connect the nodes. By establishing a set of assumptions, the problem is reduced to a shortest path problem and solved. The Dijkstra algorithm is chosen because it has only polynomial complexity, guarantees optimality and is deterministic. Graph theory and the Dijkstra algorithm are mentioned upfront because their use is pervasive in the technical literature. However, their use here is

---

<sup>1</sup>*Target rendezvous* entails the problem of multiple vehicles from separate physical locations coordinating on their arrival at a location. Although they may rendezvous in a close vicinity, in this case, they are not joining together into a formation.

in determining the optimal formation, not in determining how to maneuver a vehicle or series of vehicles into formation.

Used by Smith [31] to guide a vehicle for aerial refueling, Proportional Navigation (PN), “generates control accelerations proportional to the measured rate of rotation of the interceptor-target line-of-sight and directs these forces so as to reduce or hold constant the rotational rate of the line-of-sight [32].” The MIT Draper Laboratory [33] developed and tested a flight control system to achieve mid-air rendezvous of two UAVs. Tahk [34] proposes using only line-of-sight angles to other vehicles measured by visual sensors or radars, which requires little to no communication, makes the system more survivable to communication breakdowns, and aids in allowing a stealth mission. Tahk uses two separate guidance laws that put the vehicles on the correct path to formation rendezvous, using line-of-sight only.

Harl [10] uses Sliding Mode Terminal Guidance (SMTG) [35] to rendezvous aircraft to a final formation location because communication is not needed when using sliding mode. The sliding surfaces used are the terminal constraint, which is the final formation position for each UAV. A vector of  $m$  sliding mode vectors are established and they are “slid” to zero by defining a Lyapunov function and setting its derivative to zero. This method does not guarantee the UAVs arrive at the same time, but it does guarantee they will arrive in a finite time.

Stickney [36] describes the use of artificial potential functions may be used by identifying an objective function as an attractive force and the constraint function as a repulsive force. This allows the vehicle to be controlled in such a way that it is attracted to objectives and repulsed from constraints.

Garcia [29] performs a formation rendezvous using artificial potential functions by assigning a global variable to the manned lead vehicle which “tells” the unmanned vehicle where it should go regardless of where it begins. The first unmanned UAV to

arrive by way of an artificial potential field sink, assigns itself as the follower of the manned leader and sets a search for a follower to join its lead. This eliminates the need for the global variable and each subsequent unmanned UAV begins to join formation in a self-ordering algorithm according to the identified desired formation. The algorithm searches for a lead UAV which has a follower spot open and the algorithm continues with each UAV assigning to itself a leader and a follower until all UAVs have a “spot” in the formation. This self-ordering algorithm establishes a communication scheme that becomes valuable during other attributes of formation flight: formation hold, formation reconfiguration and splinter activities [37].

### **2.2.2 Formation Hold.**

Following successful rendezvous at the appropriate location, the challenge then turns to controlling the UAV for maintaining the formation. All of the methods identified in this section are in the outer-loop control category defined herein for sequential path planning.

Artificial potential functions are used as examples of control in both formation rendezvous and maintaining the formation. Paul [38] provides a description on the components of a potential field and how a 3-D potential field may be constructed. After using a self-ordering algorithm for formation rendezvous, Garcia [29] uses artificial potential functions to maintain the formation. A challenge with maintaining formation due to communication latency was addressed with fuzzy logic. The work herein will consider other optimal estimation techniques (Section 2.6).

Tanner [39] uses graph theory [40] to spatially relate components of a flock to one another and then used artificial potential functions as the control law. Tanner [39] begins by using a fixed topology, meaning the set of ‘neighbors’ surrounding the ‘agent’ is fixed with time. This flocking model superimposes three steering behaviors, sepa-

ration, alignment and cohesion that result in all agents moving toward a formation while avoiding collisions. Tanner [41] then uses the same methodology to maintain formation with a dynamic neighbor topology.

‘Swarming’ is an attempt at extending what is observed in nature to swarms of autonomous vehicles. Consider a flock of birds or school of fish. These animals are in formation - though no one told them to get into formation. They fly together, swim together, turn together, attain a goal together, yet there is no clear leader among them. Each of the individual entities of the flock seem to naturally relate to the members surrounding them in such a way that the group as a whole acts as one. The engineering community has attempted to describe and repeat this behavior in autonomous vehicles.

Kovacina defines emergent behavior as “the ability to accomplish complex objectives through synergistic interactions of simple reactionary components [42].” Cucker [43] uses the example of the study of emergent behavior in a flock of birds, and presents a mathematical model on the tendency of the flock to converge to a common velocity. Kovacina [42] develops a control algorithm and demonstrates the effectiveness via simulation of a swarm of vehicles to search and map a chemical cloud. In this emergent behavior control algorithm, individual agents gather information in their local environment and share that information with the rest of the swarm. As information increases, the swarm will flock to the area where help is needed to accomplish the goal. With this behavior, it is claimed that a task that is not possible for any one agent can be accomplished by a group of agents.

Often used as an inner-loop control technique, the Linear Quadratic Regulator (LQR) uses feedback to adjust the control input in order to maintain a referenced trajectory. Outside of the application to UAVs, Stankovic [11] creates a scenario of a platoon of vehicles traveling in a straight line formation as part of an integrated

vehicle highway system. LQR optimization is used to maintain a constant separation distance between the vehicles.

Venkataramanan [12] uses the MATLAB/Simulink software to simulate the inner- and outer-loop control structure for a group of UAVs flying in formation. The outer-loop controller consisted of the virtual leader's motion, the individual UAV, the individual UAV reference trajectory, and the individual UAV's controller. The inner-loop controller consisted of UAV equations of motion, engine dynamics and vortex-effect model. Results revealed robustness to swapping leader and follower and still maintaining appropriate formation separation distances.

Without prediction, a controller provides control inputs that are based on present and prior knowledge. By adding a guess or prediction on future desires, the Model Predictive Controller (MPC) [44] provides inputs based on the prediction of future states. Weihua [45] uses an MPC controller primarily to ensure collision avoidance while a formation of UAVs attempts navigation to a goal location. However, the nature of the decentralized leader/follower model may encourage the leader to avoid an obstacle and cause the formation to fall apart. To combat this, Weihua establishes a formation error term which assigned a cost to the value of maintaining the formation while navigating between obstacles on route to the final destination.

Wang [46] controls a swarm of UAVs based on the presence of obstacles. Without obstacles, the path is computed in safe mode using an outer-loop LQR path generator. While in danger mode the outer-loop path is generated using a Grossberg Neural Network [47]. Once the path has been computed, an inner-loop MPC is used for tracking the path.

Chao [48] designs a collision-free UAV formation flight controller utilizing a decentralized MPC. A cost function is designed based on velocity orientation and a distribution between UAV and obstacle avoidance to guarantee inter-vehicle collision

avoidance.

Boskovic [49, 50, 51] designs a formation-hold autopilot such that errors converge to zero asymptotically in the face of unknown leader maneuvers.

Weihua [52] applies Multiplexed Model Predictive Control (MMPC) [53] to decentralized formation flying. A typical MPC assumes all variables are updated simultaneously, which results in a high computational cost associated with multiple subsystems that are part of the formation. Utilizing the MMPC scheme, the MPC problem is solved for each subsystem (or elements of the formation) sequentially and updates the other elements of the subsystem as soon as the solution is available. The computational efficiency of using the MMPC approach allowed the UAVs to adjust their trajectory sooner and thus converge more rapidly to the final target.

### **2.2.3 Formation Reconfiguration.**

After joining formation and maintaining that formation, the mission may require a formation reconfiguration. Events such as failure of one or more communication channels, sensor failure, flight path restrictions or even full loss of aircraft are reasons a formation may need to control itself into a new configuration [13]. In addition to the control techniques reviewed in the previous sections, the artificial potential function, particle swarm optimization, and direct collocation are methods used to control formation reconfiguration.

Garcia [29] uses artificial potential functions to self-order into a new formation, just as occurred in formation rendezvous. Duan [54] uses an “improved” Particle Swarm Optimization (PSO) to find the optimal reconfiguration of multiple vehicles flying in a formation. PSO is discussed further in Section 2.4. APF and PSO are sequential outer-loop path planning techniques, but Ma [55] used an outer-loop control method known as direct orthogonal collocation to determine the optimal path to

reconfigure a formation of satellites. The direct collocation method will be discussed in the next section.

#### **2.2.4 Splinter Activities.**

Perhaps one of the most interesting aspects of formation maneuvers are the splinter activities, because they are the reason for the mission. Missions could include an ordnance drop, a battle damage assessment, target tracking, or some form of intelligence, surveillance or reconnaissance. Most of the methods discussed in this section include the outer-loop category of sequential path planning control. However, the last two methods discussed, direct and indirect methods are used to calculate a set of control inputs that provide a full outer-loop path of the optimal mission trajectory.

Swarming and emergent behavior were mentioned in the section on formation keeping, and they have relevance in performing splinter activities as well. Kovacina [42] argues that traditional control techniques do not provide the “flexibility and efficiency needed to meet the commercial and military demands placed upon UAV swarms,” and as such develops a control algorithm and demonstrates the effectiveness via simulation of a swarm of vehicles to search and map a chemical cloud. In this emergent behavior control algorithm, individual agents gather information in their local environment and share that information with the rest of the swarm. As information increases, the swarm will flock to the area where help is needed to accomplish the goal. With this behavior, a task that is not possible by any one agent can be accomplished by a group of agents.

Examples of works using emergent behavior techniques include Russell’s use of a Genetic Algorithm (GA) for UAV routing [56]. Duan [57] demonstrated the use of a PSO to control a group of UAVs to a target in minimum time, then combined the PSO and GA into a hybrid technique in which each method was run in parallel. On

each iteration, the method that provided the best solution was used in the subsequent iteration. Other biology inspired algorithms include Ant Colony Optimization [58] and Bee Colony Optimization [59], inspired by the behavior of colonies of ants and hives of bees, respectively.

The research herein led to the use of a hybrid optimization technique to include the use of a PSO algorithm to supply an initial guess to gradient-based Nonlinear Program (NLP). Additional discussion on what led to the use of a hybrid optimization technique and the PSO algorithm are discussed in Sections 2.3 and 2.4, respectively.

When considering groups or swarms of vehicles, multi-vehicle collaborative control is an area of interest in the technical community. Reynolds [60] developed three naturally observed rules for simulating flocking behavior: (1) collision avoidance, (2) velocity matching, and (3) flock centering. Various works have used these flocking behavior rules as a baseline for controlling flocks or swarms of vehicles and have varied the rules in order to solve assorted problems. Lambach [61] modifies Reynolds' rules in order to determine a flight configuration that would reduce drag and fuel consumption, which leads to increased system range and endurance. Kaiser [62] performs ISR on a point target which added nine additional rules to the three basic Reynolds' rules.

Rabbath [63] offers a review of several papers covering the topic, including Chandler [64] who rendezvoused three UAVs from physically separate locations to a final location at the same time. Each vehicle plans its own path with an *approximate* optimal path that is refined according to maneuver constraints. A sensitivity function, communicated from each UAV, is calculated by a coordinating agent. The coordinating agent broadcasts the optimal time of arrival for all UAVs. The UAVs then adjust their own path according to the broadcast time [65].

Shima [66] establishes a scenario in which a group of micro-UAVs are released from

a small UAV in order to investigate targets in an urban environment. The micro-UAVs know the targets, but upon release have no coordination or collaboration on how to attain this mission. Multi-vehicle collaborative control gathers information on the targets and available micro-vehicles and assigns a target to each micro-UAV according to a computed optimal path for each vehicle to accomplish the overall mission.

Shima [66] and Chandler [64] utilize a centralized planner to gather information and send out tasks to the individual UAVs. Ryan [67] utilizes a decentralized structure to control a group of UAVs to patrol an area searching for intruders while avoiding constraints in the grid space. These vehicles collaborate in real-time based on shared information and their own local information.

Alighanbari [68] uses Robust Decentralized Task Assignment (RDTA) [69] in which a solution is determined in two phases. In phase 1, each vehicle computes a set of *candidate* scenarios for itself and sends the information out. This, in essence, is an initial guess for the next phase. In phase 2, each vehicle computes its own scenario based on the set of candidate scenarios. This cuts down the computation time and high volume of communication data significantly. This method is reliant on a good set of candidate scenarios computed in the first phase.

Garcia [29] uses artificial potential functions to control a splinter activity in which three UAVs are commanded to ‘hover’ around a hot-spot until their manned lead is deemed a safe distance from the hot-spot, then rejoin the formation.

The next set of techniques are gradient-based numerical methods for solving optimal control problems (discussed further in Section 2.3). Furukawa [70] uses indirect methods to minimize time to a final formation. In the indirect method, the cost, constraints and boundary conditions are established. The cost function is then augmented with the constraints and boundary conditions using Lagrange multipliers.

Using the Calculus of Variations, a set of first-order necessary conditions for optimality are established [71]. These first-order necessary conditions can be discretized and solved using a Nonlinear Program (NLP) solver such as MATLABs *fmincon* in order to obtain costates, optimal control and the associated optimal states.

Jorris [72], Masternak [73], and Smith [8] used direct orthogonal collocation as a method for transcribing an optimal control problem into an NLP. This method approximates the states and controls with interpolating polynomials, where in order to minimize the interpolation error between the points, the interpolation conditions are enforced at the roots or extrema of orthogonal polynomials. These approximations are substituted into the optimal control problem directly, rather than the necessary conditions for optimality, resulting in an NLP. Using commercially available solvers such as SNOPT [74] or IPOPT [75], the NLP is solved using a gradient-based search method like sequential quadratic programming (SQP) or by using an interior point method. More on the use of direct orthogonal collocation techniques for solving optimal control problems can be found in Benson [76] and Huntington [77].

Jorris' [72] candidate scenario for the global strike problem was similar to the loyal wingman scenario in which a UAV begins with a fixed set of boundary conditions and traverses intermediate waypoints on route to a final destination. No-fly zones or path constraints were established which the vehicle must avoid and multiple cost functions were evaluated, including minimize time and minimize control. Masternak [73] later solved a modified version of the global strike scenario including multiple objectives, no-fly or keep out zones, and he proposed direct collocation as a method for solving the optimal control problem.

Smith's [8] scenario proposed a UAV, operating in the National Airspace, flying a precomputed mission path, when sensors detect an intruder aircraft. The UAV must use the sensor information to estimate the intruder's future state, model the future

state as a keep-out zone and successfully avoid the aircraft while returning to the original flight path, optimized with respect to various objective functions. Smith’s scenario must account for dynamic, non-deterministic intruders that “pop-up” during execution of the precomputed mission plan, similar to the loyal wingman candidate scenario *as defined herein*.

### **2.2.5 Research Approach.**

The previous four subsections established multiple methods for controlling UAVs and groups of UAVs throughout the four attributes of formation flight. The loyal wingman seeks an outer-loop control technique to compute an initial optimal mission path as well as computing a trajectory re-plan due to changes in the mission environment. Therefore, outer-loop control techniques which continually define a path based on high frequency tracking updates will not be utilized.

In distinguishing between direct and indirect methods for solving the optimal control problem, the use of direct orthogonal collocation in combination with the GPOPS II [78] MATLAB® [79] based software allows for rapid solutions to optimal control problems, which addresses the near real-time optimal control that is necessary in the loyal wingman candidate scenario. The similarities of the loyal wingman scenario to works discussed in Section 2.2.4 reveal the usefulness of this method. Therefore, this work proposes the use of the rapid and accurate direct orthogonal collocation method to solve the loyal wingman optimal control problem *as defined herein* as a unique and original contribution to the scientific and engineering body of knowledge.

## **2.3 Nonconvex Optimal Control**

Kirk [71] provides a definition of a convex set and a method to check the convexity of a function. If the function is convex, then a solution to the optimal control problem

is guaranteed to be the globally optimal solution. Unfortunately, many optimal control problems of interest are nonlinear and the solution therefore is considered only locally optimal.

Highly nonlinear, nonconvex problems have a large number of locally optimal solutions. The challenge is to use a solution method that is most likely to converge on the best locally optimal solution. The direct orthogonal collocation method chosen to solve the loyal wingman problem uses a gradient-based numerical method for solving the NLP, which requires a user supplied initial guess. Jodeh [80] used a straight line trajectory as an initial guess with only a fraction of the desired discretized points, then used the low-fidelity solution as an initial guess into a higher-fidelity model. Smith [8] and Suplisson [81] have a continually updating solution where previous solutions are used as an initial guess into the new update, but still required a method to determine the first solution.

There is a tendency for NLP optimizers to converge in the region of the initial guess [82], therefore, a method is desired for supplying an initial guess for the loyal wingman solution that is in the region of the best locally optimal solution.

Rao [83] broadly categorized numerical methods for solving optimal control problems into gradient and heuristic methods. Gradient methods utilize derivative information provided in the problem formulation to search deterministically for the optimal solution. Gradient methods are local optimization methods, meaning the results converge to locally optimal solutions. Conversely, heuristic methods begin with a set of possible solutions and use a stochastic search to continually update and adapt the initial set of solutions until an optimal solution is found. Contrary to the gradient-based methods which converge to locally optimal solutions, heuristic methods are a global technique. Throughout the work herein when using the phrase global region or global nature of the PSO, this refers to the good possibility that the PSO

provides to move toward the best locally optimal solution within the space that is searched, without guarantee that the PSO will converge. Examples of heuristic methods include genetic algorithms, particle swarm optimization, ant colony optimization [84], differential evolution [85] and simulated annealing [86].

Conway’s [87] survey suggested the best method for solving optimal control problems was either heuristic algorithms alone or heuristic algorithms in combination with transcription techniques which utilize the gradient-based NLP. Englander [88] and Chilan [89] used a heuristic technique to generate the outer-loop solution. Englander continued to use a heuristic technique to generate the inner-loop solution, while Chilan used a gradient-based technique for the inner-loop solution. Showalter [90] and Vinko [91] used hybrid heuristic optimal control techniques for space-based applications. Modares [92] performed an experiment comparing various hybrid techniques and concluded that a modified heuristic particle swarm optimization combined with a gradient-based sequential quadratic program is robust and accurate when compared with other methods. Additional works [82, 93, 94, 95, 96] have used a hybrid PSO with gradient-based NLP.

The work herein applies a hybrid optimization technique that exploits the speed and global nature of a heuristic PSO to generate an initial guess to a gradient-based NLP, which exploits the speed and accuracy of direct orthogonal collocation (DOC) to provide a rapid, feasible solution to the loyal wingman intermediate-target optimal control problem.

## **2.4 The Particle Swarm Optimization**

### **2.4.1 Particle Swarm Optimization Background.**

The Particle Swarm Optimization (PSO) was introduced by Kennedy and Eberhart [97]. Based on the behavior of flocks (or swarms) of birds, each particle, which

is in general any scalar or vector quantity the user desires, is flown through space in search of the optimal solution with respect to a defined fitness or cost function. The basic PSO algorithm is two lines of code which first updates its velocity (the rate at which a solution changes) based on its previous velocity (often termed the ‘inertial’ component) and the behavior of the flock (or swarm); and then uses the updated velocity to update the particle’s position<sup>2</sup>. The algorithm is seeded with an initial particle position and velocity and iterates through the following two formulas to update the particle’s velocity,  $v(i)$ , and position,  $x(i)$ ,

$$v(i+1) = a_1v(i) + b_1r_1(x(i) - p_L) + b_2r_2(x(i) - p_G) \quad (2.1a)$$

$$x(i+1) = x(i) + v(i+1). \quad (2.1b)$$

The updated velocity, Equation 2.1a, of an individual particle is achieved through a three-part formula. The inertial component provides the weighting,  $a_1$  for the particle’s previously computed velocity. The first ‘social’ component provides weighting,  $b_1$ , on the best solution found so far for each individual particle. On any given iteration, if the cost with  $v(i)$  and  $x(i)$  is better than that for  $p_L$ , then that particle’s ‘local best’ is updated. The second social component,  $b_2$ , provides a weighting on the best solution found for the entire ‘swarm’ (all particles). On any given iteration, if a better solution is found for the swarm, then the global best is updated. Each of the social components have a deterministic parameter weighting ( $b_1$  and  $b_2$ ) as well as a nondeterministic evenly distributed parameter weighting,  $r_1, r_2 \in [0, 1]$ .

The basic PSO algorithm is only two lines of code, but a review of literature reveals there are many methods for varying the PSO algorithm for individual applications.

---

<sup>2</sup>‘position’ and ‘velocity’ when referenced to discussion on PSO refer to the current value of the particle’s position and the rate the the particle’s position is changing

### 2.4.2 PSO Parameters.

Convergence characteristics of the PSO algorithm are dependent both on the application as well as the choice of parameters. Based on a mathematically complex work done by Clerc and Kennedy [98], Trelea performed an experiment revealing these convergence tendencies [99]. Clerc [100] suggested scaling the update to velocity by a constriction factor,  $K$ , as necessary to ensure convergence. The use of a constriction factor weights the inertial component, eliminating the need to identify a weighting ( $a_1$ ) for the inertial component. Equations 2.1a and 2.1b are updated with a constriction in Equation 2.2.

$$v(i+1) = K[v(i) + b_1r_1(x(i) - p_l) + b_2r_2(x(i) - p_g)] \quad (2.2)$$

The formula for the constriction factor can be mathematically complex such as the one proposed by Lu [101] which allows trigonometric variation of the constriction factor such that a higher constriction during early iterations allows for a more global search, while a smaller constriction value in later iterations focused search in the local regions. Clerc [100] recommended a simplified version and will be used in the loyal wingman application,

$$K = \frac{2}{\phi - \sqrt{\phi^2 - 4\phi}}, \quad (2.3)$$

with  $\phi = b_1 + b_2$ ,  $\phi > 4$ . Eberhart and Shi [102] determined that the constriction factor should be used in combination with a max value assigned to velocity,  $v_{max} = x_{max}$ , so  $v$  is bound by  $x_{max}$  at each iteration.

### 2.4.3 PSO Seeds.

All works studied used random methods for producing the initial particles [54, 57, 93, 94, 103, 104, 105, 106, 107, 108] to seed the algorithm.

#### **2.4.4 PSO Constraints.**

Hu and Eberhart [109] suggest initializing the algorithm with a set of seeds that meet constraints, then at each iteration only keeping particles that continue to meet constraints. Pontani [105] suggests that any particle that does not meet constraints at each iteration should have its velocity component set to 0, essentially throwing out that iteration. Mazhoud [107] provides a review of methods for handling constraints, including penalty functions [110], self-adaptive velocity [111] and stagnation determination [112].

#### **2.4.5 PSO Model Order.**

Duan [54, 106, 57] used the PSO to choose the elements of a vector that provide the optimal solution with respect to the established cost function. Others, such as Conway suggest computation time efficiency by modeling the control input as a polynomial and using the PSO to solve for the polynomial coefficients. Zhuang [93] shows that his problem has a property of differential flatness which allows a reduction in the number of variables.

#### **2.4.6 PSO Conclusion.**

The work herein will use these ideas to tailor a PSO algorithm to the loyal wingman scenario.

### **2.5 Models**

The literature review to this point has focused on determining a method for solving the optimal control problem. The purpose of this section is to present various aspects of the problem that must be modeled, covering threats, UAV dynamics, and multiple-target (waypoint) missions.

### 2.5.1 Loyal Wingman.

A state model must be determined for the loyal wingman that will be used as dynamic constraints in the optimal control problem. Carter [7] chose a high-fidelity 12-state, 6 Degree-of-Freedom (DOF) model because of the highly dynamic and critical maneuvering capabilities necessary in high performance fighter air vehicles. This high-fidelity model is computationally expensive, but is critical to obtaining accurate results. If the fidelity were not necessary, then it would be resourceful to utilize a reduced-order model to increase computational efficiency. Smith chose a 5-state, 3-D, 2-control model consistent with the AFRL program document [113] where the controls are vertical acceleration,  $N_z$ , and bank angle,  $\mu$ . The work herein uses the same model, but in order to avoid ‘bang-bang’ control, Smith’s controls are made states and their rates are made the controls,

$$\begin{aligned}
\dot{x}(t) &= V \cos \gamma(t) \cos \chi(t) \\
\dot{y}(t) &= V \cos \gamma(t) \sin \chi(t) \\
\dot{z}(t) &= V \sin \gamma(t) \\
\dot{\gamma}(t) &= \frac{N_z g \cos \mu(t) - g \cos \gamma(t)}{V} \\
\dot{\chi}(t) &= \frac{N_z g \sin \mu(t)}{V \cos \gamma(t)} \\
\dot{N}_z(t) &= u_1(t) \\
\dot{\mu}(t) &= u_2(t),
\end{aligned} \tag{2.4}$$

where  $x, y, z$  represent position coordinates,  $\gamma$  is flight path angle,  $\chi$  is heading,  $N_z$  is vertical acceleration,  $\mu$  is bank angle, controls are vertical acceleration rate,  $u_1 = \dot{N}_z$  and bank angle rate,  $u_2 = \dot{\mu}$ . Velocity,  $V$  is assumed constant and  $g$  is the gravitational constant.

### **2.5.2 Threats.**

The loyal wingman will operate in a hostile environment where, if the loyal wingman is detected, the chances are high that the mission will not be completed. This section reviews how other literary works model threats.

#### **2.5.2.1 Static Threat Models.**

Garcia [29] uses artificial potential functions as a means of controlling the UAVs. The obstacles were similarly modeled using vector fields with weighted sigmoid functions, which works well for static and predictably slow-moving obstacles. Jorris models two no-fly zones as cylinders that extended infinitely high [72]. Weihua [45, 114] uses simple circles to model small obstacles, but if the obstacle is determined large, then Weihua uses polytopes [115] to model the obstacles, because of the variety of shapes that can be generated. Given a chosen shape, a hexagon for example, each of the 6 edges are assigned a 1 for its safe side and a 0 for its unsafe side. Each discretized point of the full trajectory is evaluated against each edge line of the shape to determine whether the trajectory is on the “safe” or “unsafe” side of the line. For an individual discretized point of the trajectory, its binary “safe” and “unsafe” values are summed and if the value is greater than or equal to 1, then the threat is successfully avoided at that point in the trajectory.

#### **2.5.2.2 Modeling a Changing Threat Region.**

Campbell [116] models dynamic, non-deterministic threats by first modeling a constraint as an ellipsoid then adapting the ellipsoid by developing an extended set membership filter. This filter linearizes the nonlinear model at each discrete time step, then adds the remainder as uncertainty, delivering an ellipsoid set. Campbell further tightens the estimate and approximates the intersection of the two ellipsoid

estimates as a polyhedron. A final ellipsoid bounds the polyhedron and is used as the obstacle that must be avoided.

Smith [8] uses a particle filter to estimate future location of a dynamic intruder vehicle for avoidance. The obstacle's particle distribution is modeled using a Minimum Volume Enclosing Ellipsoid (MVEE) [117, 118] and projected along a trajectory. The set of ellipsoids created by the particle filter and MVEE algorithm are then interpolated into a single keep-out region using the SLIMVEE [8] algorithm, adapted from Schoemake's Slerp [119] algorithm.

Smith additionally used superquadrics as described by Barr [14]. The product of a half ellipsoid in the  $(x, y)$  plane with the full ellipsoid in the plane orthogonal to the  $(x, y)$  plane can be written as

$$\begin{bmatrix} a_1 \cos^{\varepsilon_1}(\eta) & \cos^{\varepsilon_2}(\omega) \\ a_2 \cos^{\varepsilon_1}(\eta) & \sin^{\varepsilon_2}(\omega) \\ a_3 \sin^{\varepsilon_1}(\eta) \end{bmatrix}, \quad (2.5)$$

where

$$-\pi/2 < \eta < \pi/2 \quad (2.6a)$$

$$-\pi < \omega < \pi. \quad (2.6b)$$

An implicit equation can be derived from the above Equation 2.5

$$F(x, y, z, a_1, a_2, a_3, \varepsilon_1, \varepsilon_2) = \left( \left( \frac{x}{a_1} \right)^{\frac{2}{\varepsilon_2}} + \left( \frac{y}{a_2} \right)^{\frac{2}{\varepsilon_2}} \right)^{\frac{\varepsilon_2}{\varepsilon_1}} + \left( \frac{z}{a_3} \right)^{\frac{2}{\varepsilon_1}} \quad (2.7)$$

where  $a_i$  is a principal axis,  $\varepsilon_1$  determines the shape of the superellipsoid cross section in a plane perpendicular to the  $(x, y)$  plane and  $\varepsilon_2$  determines the shape of the superellipsoid cross section parallel to the  $(x, y)$  plane. Equation 2.7 is the general

formula for modeling a class of superquadrics, called the inside-outside function. For a given point  $(x, y, z)$ , if  $F < 1$ , the point is in the interior of the geometric shape, if  $F > 1$ , the point is exterior to the geometric shape and if  $F = 1$ , the point is on the surface of the geometric shape. By varying the principal axes values  $a_i$  as well as the  $\varepsilon_i$  values, various superquadric shapes may be generated such as spheres, superellipsoids, and cylinders [120]. Smith varied the equation slightly to use only a single shaping parameter to model spheres, ellipsoids and cylindrical superquadrics,

$$F(x, y, z, a_1, a_2, a_3, \varepsilon) = \left(\frac{x}{a_1}\right)^\varepsilon + \left(\frac{y}{a_2}\right)^2 + \left(\frac{z}{a_3}\right)^2. \quad (2.8)$$

### 2.5.2.3 Minimize Threat Exposure.

The desire is not always a strict keep-out region, but rather to minimize exposure. Gaunt [121] calculated the Signal to Noise Ratio (SNR) and then developed a formulation that can be used inside a cost functional to minimize exposure. His formulation forces the cost functional to equal 1 when the aircraft is at the radar detection limit and then grow quickly as the vehicle traverses deeper into the threat area. This method allows the threat areas to be modeled in the cost function as a continuous, differentiable, conditional constraint. Jodeh [80] uses a similar technique, modeling a conditional constraint in the cost function with the use of a sigmoid or barrier function. Smith [8] used a sigmoid function as well to establish an inequality constraint in his Sense and Avoid (SAA) optimal control problem. The keep-out zone for two aircraft approaching one another is the Federal Aviation Administration (FAA) required minimum vertical and horizontal separation distances specified for the National Air Space (NAS). The need to avoid both a vertical and horizontal separation distance translates to the need to avoid multiple threats. Smith evaluated the difference between summing the sigmoid of multiple threat zones and taking the

product of the sigmoid of multiple threat zones and determines the sigmoid product method as superior [8].

#### 2.5.2.4 Dynamic, Non-Deterministic Threat Models.

Threats may include air or ground threats. Li and Jilkov [122] survey various dynamics by which to model maneuvering intruder aircraft. By determining the appropriate assumptions, such as whether the vehicle is maneuvering or non-maneuvering, aggressive or non-aggressive, a 2-D or a 3-D dynamics model, and the way noise is modeled in the system, there are various dynamic models from which to choose. Smith [8] chose a Singer [123] acceleration model, which assumes non-aggressive, uncoordinated, maneuvering aircraft with time auto-correlation,  $\alpha$ .

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\alpha \end{bmatrix} \quad (2.9)$$

Stankovic [11] develops a single axis model to control a platoon of vehicles. Assumptions in the loyal wingman problem will result in a simple linear model that combines these works.

#### 2.5.2.5 Sensor Measurement Updates.

Maroney [124] investigates the concept that there may be a variety of sensor combinations that are appropriate for various UAVs according to their capability. Chen [125] integrates the most appropriate set of sensors for the sense and avoid mission application. The work herein assumes that the appropriate sensor and measurement integration techniques are available for the loyal wingman to appropriately track and avoid threats in the air and on the ground.

### 2.5.2.6 Possible Threats.

The research sponsor provided a list of threats that may be applicable to the loyal wingman problem. Table 2.5.2.6 lists threats and information necessary to model them as superquadrics. In addition, the 3-D contour of the superquadric threat keep-out regions are represented in Figure 2.1. These images are provided here because graphical results of threats modeled in 3-D in subsequent chapters are displayed on a 2-D image where it may be difficult to see a loyal wingman trajectory that avoids the threat using the altitude dimension.

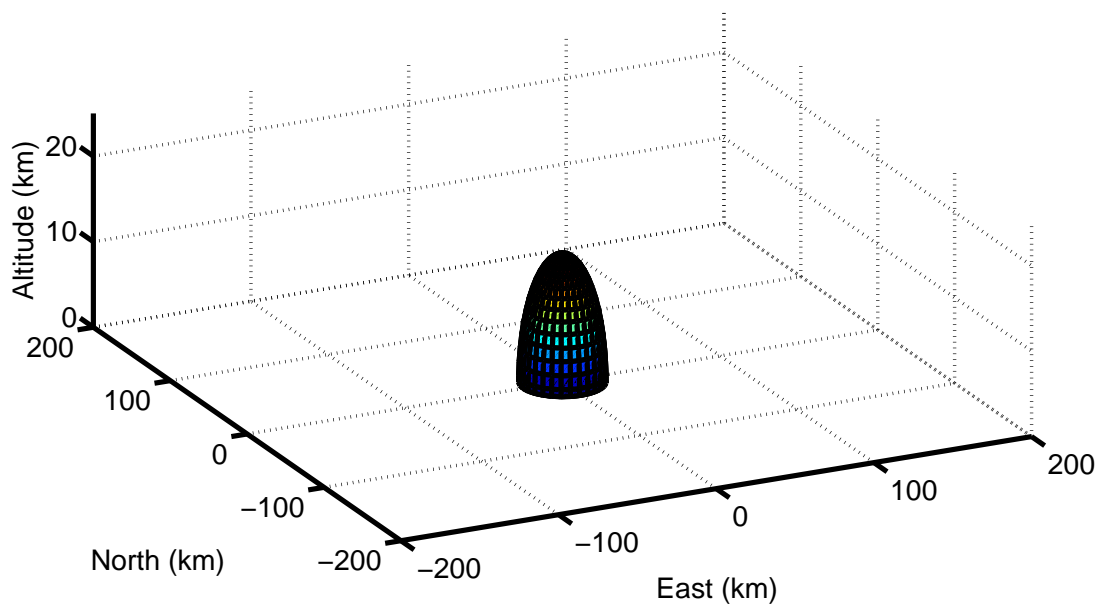
**Table 2.1. Possible Loyal Wingman Threats**

Type	Range (km)	Altitude (km)	3D Shape
Surface to Air Missile (SAM)	25	25	Sphere
Anti-Aircraft Artillery (AAA)	18	15	Ellipsoid
Networked Radar	180	15	Ellipsoid
Manpad	6	3	Ellipsoid
Electro-Optical Sensor	25	15	Ellipsoid
Weather	20	15	Cylinder
Threat on Road	100x20	20	Cylinder

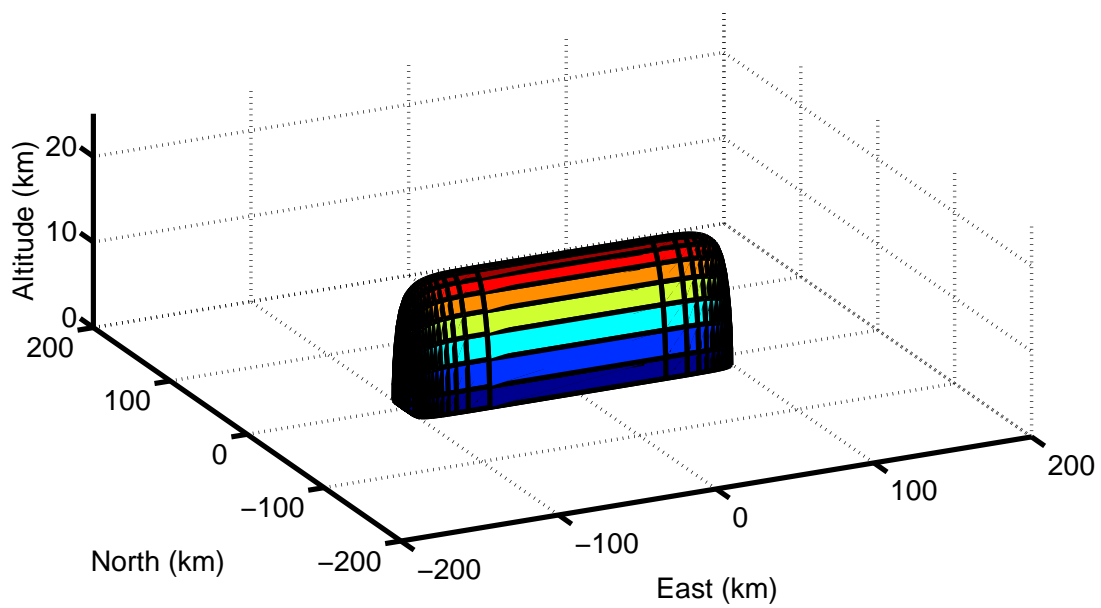
### 2.5.3 Multiple-Target Mission.

The loyal wingman candidate scenario identifies the requirement to rendezvous at multiple mission targets on route to a fixed final time and location. Beard [126] models threats as Voronoi polygons which must be avoided, then identifies waypoints at the corners of the polygons. The path planner then traverses the edges of the polygon from one waypoint to another. Lee [127] similarly has a UAV track a ground vehicle using waypoint path planning by establishing different algorithms based on the relative speed of the UAV and ground vehicle.

Benson specifies that there are a number of conditions for which the direct orthogonal collocation method is best solved using multiple phases, one of which is an



(a) EO Sensor, 25km Range by 15km Altitude



(b) Estimated Threat Region Extended Along Road

Figure 2.1. 3-D Threat Keep-Out Regions Using Superquadrics

interior point constraint such as a waypoint [128]. Jorris [72], using direct orthogonal collocation, showed the need to put a phase break at the constraint.

## 2.6 Stochastic Estimation

The current chapter proposed both optimal control and stochastic estimation techniques to solve the loyal wingman problem *as defined herein*. This section examines techniques from the literature that may be used to estimate moving, non-deterministic threats. The choice of estimation tool will be dependent on assumptions and user desires, including linearity of the model, estimation probability distribution, and computational efficiency.

### 2.6.1 Kalman Filter.

The Kalman filter was first introduced in a paper by R.E. Kalman in 1960 [129]. Around the same time of the introduction of this discrete-data linear filter technique, advances in digital computer technology made it possible to implement real-time recursive estimating solutions. Brown and Hwang provide an example Kalman filter algorithm [130]. The key attributes of the Kalman filter are the use of a linear model for propagating states and the assumption that the probability distribution of the estimate is Gaussian. Additionally, the simplicity of the Kalman filter means the computational complexity is low when compared to particle based methods such as the unscented Kalman filter and the particle filter.

Shima and Rasmussen [131] demonstrate a control technique for multiple vehicle collaborative control using the Kalman filter. In their paper, each UAV runs multiple filters in parallel estimating its own states as well as running filters on its own states as viewed by other members of the group. Singer [123] demonstrates the use of the Kalman filter for the specific purpose of tracking maneuvering targets. This method

allows rapid, *a priori* estimates of tracking performance by sensors providing any combination of range, bearing, and elevation measurements.

### 2.6.2 Extended Kalman Filter.

The Extended Kalman Filter (EKF) is a variation of the Kalman filter that allows for recursive estimation of measurement and state propagation functions that are nonlinear. The EKF linearizes the measurement and state propagation functions about the estimated trajectory by taking the partial derivatives of the measurement and state propagation functions, which can then be used in the recursive Kalman filter equations. Although this may seem convenient, there are risks. First, large values in the original covariance matrix combined with low-noise measurements at the first step will cause the covariance matrix to jump from the original large value to a very small value. Roundoff errors associated with computation of the covariance matrix can cause numerical computing issues that must be dealt with appropriately. A potential solution is to ensure symmetry and positive definiteness of the covariance matrix on the first step [130].

The second issue is the error associated with the linearization of a nonlinear function, which takes the first two terms of the Taylor series expansion. In some cases, the linearized matrix is not an accurate representation of the original function, which means there is an error from the beginning that is then propagated indefinitely through the model, causing the error to continually grow and in some cases causes divergence. In choosing the EKF as an estimating tool, care must be taken to ensure these potential problems don't cause inaccurate estimates for the problem one is trying to solve [130]. Smith [8] compared the various filters for a collision avoidance estimation problem. His results showed an example where reduced observability magnified the linearization error of the EKF.

### 2.6.3 Unscented Kalman Filter.

Wan [132] introduces the Unscented Kalman Filter (UKF) as an alternative to the EKF which addresses the suboptimal performance associated with the linearization of the state propagation and measurement update functions. The UKF is accurate to the third order for Gaussian inputs for all nonlinearities and accurate to the second order for non-Gaussian inputs. Additionally, the UKF computational complexity is of the same order as the EKF. For this reason, Wan [132] describes the UKF as clearly superior to the EKF. Ross [133] simultaneously solved the optimal control problem and the optimal estimation problem for a bearing-only sensor using the UKF. Smith [8] also performed a comparison of various models and showed results where a UKF clearly outperformed the EKF. For instances where the state model is nonlinear and posterior distribution estimates can be assumed a Gaussian and Normal distribution, it is clear the appropriate estimation technique is the UKF.

### 2.6.4 Particle Filter.

The UKF and EKF were presented as options for estimation when measurement or state propagation functions are nonlinear. These techniques still assume a Gaussian posterior probability distribution. In cases where the posterior probability distribution cannot be assumed Gaussian, another estimation technique is needed. The particle filter operates by first creating a set of particles,  $\chi$ . These particles are then propagated through the state transition function. Because there is a non-deterministic input value associated with the state transition matrix, the particles that propagate through the state transition matrix all have varying values. This propagation continues until there is a measurement update. Each of the particles are then assigned a weighting based on a likelihood function. This function allows each particle to evaluate its likelihood of being true based on the value obtained from the measurement.

The probability distribution function is then found by normalizing the weights of the particles. An algorithm for the particle filter provided by Kim [134] and examples of variations of the particle filter and methods and algorithms for re-sampling are provided by Arulampalam [135].

The benefit of particle filters is they are robust to handle nonlinear models and there need not be Gaussian assumptions on the posterior probability distribution. However, the expense is typically a high computation cost based on the number of particles that are used in the filter. There are a number of examples of use of particle filters for the control of UAVs in literature. Conde [136] uses a particle filter to predict the trajectory of a UAV and in particular highlights the importance of re-sampling in the results of the work. Gustafsson [137] demonstrates a technique known as Rao-Blackwellization by using a combination of a particle filter and a Kalman filter for estimation. He showed that for a high-dimensioned, 27-state problem, 24 of the states were linear and could be estimated using the Kalman filter. The particle filter was then used to estimate the three remaining position states. Combining the use of the filters assured no loss of fidelity while significantly reducing the computational cost.

## 2.7 Trajectory Re-Planning

The final aspect of the loyal wingman optimal control problem requires a review of techniques that allow for dynamic, near-real-time trajectory re-planning. In Smith's [8] SAA scenario, a UAV operating in the NAS is flying a precomputed path. A suite of sensors provide information on other potential aircraft that must be avoided. A receding horizon model predictive controller continually projects an image of the airspace a given time in the future,  $t_h$ . If necessary, a path is created which optimally avoids intruder aircraft and returns to the previously identified path. The aircraft then flies the newly computed path for a portion of the horizon  $t_p$  and

sequentially updates the collision avoidance trajectory every  $t_p$ . An inner-loop controller such as an LQR is used to ‘fly’ each  $t_p$ . There are challenges with using an LQR [138] and so Jiang [139] uses Neighboring Optimal Control (NOC) to obtain a solution to return to a nominal path. Described further by Yan [138], NOC establishes a set of perturbation equations linearized around the nominal trajectory, assumes the solution in the form of polynomials discretized at the roots of an orthogonal basis set and may then quickly find the solution to a set of algebraic equations. NOC uses the same direct orthogonal collocation transcription technique used in the loyal wingman optimal control problem described herein and therefore may be a convenient tool for dynamically re-planning a mission.

Ground-based entities include input from manned operators as well as the use of computer subsystems such as the Operations Research Planning and Utility System (OPUS) [140], described as an A-star heuristic search algorithm for very fast computation [140]. The Mission Reconfigurable Cockpit (MRC) Real-time In-flight Planner [141] is an onboard dynamical re-planning tool which utilizes digital terrain elevation data (DTED) to create a small set of potential solutions and chooses the optimal of the small set as its solution. The benefit of this method is it develops plausible new trajectories in near real-time.

Both NOC and the receding horizon model predictive controller are a few of the many tools available for inner-loop trajectory tracking or sequential trajectory creation as described in the literature [142]. These tools, however assume an outer-loop solution has already been created and solves the problem of either maintaining that path or avoiding disturbances/threats that may traverse the path and require continual use of computation resources that are assumed not continuously available throughout the mission. OPUS and the MRC planner both provide full outer-loop path re-plans, however the ground-based OPUS requires a human in the loop and

a high communication bandwidth which is not suitable for the low-communication bandwidth requirement of the loyal wingman optimal control problem, whereas the MRC planner is suboptimal.

The work herein assumes inner-loop control and sequential path-planning techniques such as the ones mentioned previously are available. The changes in the loyal wingman mission environment are significant enough to require a full mission re-plan in near real-time. This research focuses on the use of optimal control techniques to rapidly create both an initial optimal outer-loop mission path as well as to re-plan the path in a changing mission environment.

## **2.8 Literature Review Conclusion**

Chapter II began with a review of literature related to MUM-T and loyal wingman applications and concluded that solving the loyal wingman problem as defined in Chapter I with optimal control and stochastic estimation techniques will provide valuable contributions to the scientific and engineering community. A review was conducted across the four attributes of formation flight and determined direct orthogonal collocation as a rapid and accurate optimal solution technique. A discussion then ensued on the locally optimal nature of gradient-based methods and suggested the use of a hybrid optimization technique in which a particle swarm optimization algorithm generates a solution in the correct global region and is then provided as an initial guess to the gradient-based NLP optimizer. Having determined the appropriate hybrid optimal control methodology, a review was then conducted on various models that are used throughout literature for modeling UAV dynamics as well as various ways to model threats. The literature review concluded with a discussion on stochastic estimation techniques and methods used for near real-time trajectory re-planning.

The next two chapters will define the optimal control methodology.

### III. Formulate the Optimal Control Problem

The purpose of this chapter is to answer research question one by formulating the optimal control problem and demonstrating the direct orthogonal collocation technique using a reduced-order 3-state, 2-D model. After formulating the components of the optimal control problem in Section 3.2, Section 3.3 provides a concise problem formulation summary. Results in this chapter highlight the importance of providing a good initial guess to the gradient-based NLP and the conclusion highlights contributions that result from formulating the optimal control problem. The usefulness of a hybrid optimal control methodology, which utilizes the PSO to provide an initial guess to the NLP, is demonstrated in the next chapter.

#### 3.1 The Optimal Control Problem

The goal in an optimal control problem is to find an admissible control,

$$\mathbf{u}(\mathbf{x}(t), t) \tag{3.1}$$

that minimizes an identified objective or cost functional

$$J(\mathbf{x}(t)) = \mathbf{h}(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) dt \tag{3.2}$$

where

$$\mathbf{h}(\mathbf{x}(t_f), t_f) \tag{3.3}$$

is a term relating to the cost at the final state (terminal cost) and

$$\int_{t_0}^{t_f} \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) dt \tag{3.4}$$

is a term relating to the cost along the path (running cost). The solution must also satisfy dynamic path constraints of the form,

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t), t), \quad (3.5)$$

inequality and equality constraints along the path (no-fly zones) of the form,

$$\mathbf{c}(\mathbf{x}(t), \mathbf{u}(t), t) \leq \mathbf{0}, \quad (3.6)$$

and boundary conditions of the form,

$$\mathbf{x}(t_0) = \mathbf{x}_0 \quad (3.7a)$$

$$\mathbf{x}(t_f) = \mathbf{x}_f. \quad (3.7b)$$

The components of the optimal control problem formulation will now be described for the loyal wingman problem.

### 3.2 Components of the Loyal Wingman Problem Formulation

The loyal wingman candidate scenario involves a UAV that must autonomously compute and fly to an intermediate target (waypoint) on route to a final rendezvous, while either completely avoiding threats or minimizing the risk of exposure when threats are not avoidable. The purpose of this section is to formulate the optimal control problem in a way that will account for the various threat and boundary conditions scenarios.

### 3.2.1 Dynamic Constraints.

The 3-state, 2-D, point-mass model, which does not change throughout the mission is

$$\dot{x} = V \cos \psi \quad (3.8a)$$

$$\dot{y} = V \sin \psi \quad (3.8b)$$

$$\dot{\psi} = u, \quad (3.8c)$$

where  $x$  and  $y$  are the UAV's  $x$ - and  $y$ - positions,  $\psi$  is heading,  $u$  is the control, heading rate in  $rad/s$  and  $V$  is the velocity, held constant. The 3-state vector,  $\{x, y, \psi\}$ , will be represented as  $\mathbf{x}$ .

### 3.2.2 Boundary Conditions.

The boundary conditions include three final condition scenarios:

1. Fixed final time,  $t_{fc}$ , to fixed final point,  $\mathbf{x}_{fc}$ ,
2. Minimize time to fixed final point,  $\mathbf{x}_{fc}$ , and
3. Minimize time to rendezvous with the manned lead,

where the subscript  $fc$  indicates the final condition. Two assumptions are established for rendezvous with the manned lead. First, the time-dependent path of the manned lead is known *a priori*. Future work may consider stochastics in the manned lead's trajectory and the techniques discussed in Chapter VI may be extended for use in that scenario. Second, it is assumed the loyal wingman needs to arrive within a safe distance of the manned lead and other systems or algorithms will ensure a collision-free final formation rendezvous. For simplicity, the optimal control problem is feasible for a subset of the general problem when the states at the final time,  $t_f$ , of the loyal

wingman, subscript  $W$ , are equivalent to the states at the final time of the manned lead, subscript  $L$ , such that

$$\mathbf{x}_W(t_f) = \mathbf{x}_L(t_f). \quad (3.9)$$

In order to satisfy intermediate target constraints, the problem is broken into multiple phases, one break for each waypoint, a common technique when using direct orthogonal collocation to solve optimal control problems. The loyal wingman begins at the established initial conditions.

Phase 1 initial condition (superscripts indicate phase number):

$$\mathbf{x}_W^1(t_0) = \mathbf{x}_0, \quad (3.10)$$

where in the case of the manned rendezvous  $\mathbf{x}_0 = \mathbf{x}_L(t_0)$ .

Phase 1 final condition:

$$\begin{aligned} x_W^1(t_f^1) &= x_1 \\ y_W^1(t_f^1) &= y_1. \end{aligned} \quad (3.11)$$

Phase 2 initial condition is at the intermediate target location (the superscript 2 indicates phase number):

$$\begin{aligned} x_W^2(t_0^2) &= x_1 \\ y_W^2(t_0^2) &= y_1. \end{aligned} \quad (3.12)$$

Phase 2 final condition is different for each scenario

1. Fixed final time to fixed final point,  $\mathbf{x}_W^2(t_f^2) = \mathbf{x}_{fc}$ ,  $t_f^2 = t_{fc}$
2. Minimize time to fixed final point,  $\mathbf{x}_W^2(t_f^2) = \mathbf{x}_{fc}$
3. Minimize time to rendezvous with the manned lead,  $\mathbf{x}_W^2(t_f^2) = \mathbf{x}_L(t_f^2)$

Finally, in a multiple-phase problem, a linkage constraint must be established that ensures the final time and states of phase 1 are exactly equal to the starting time and

states of phase 2,

$$\mathbf{x}_W^1(t_f^1) = \mathbf{x}_W^2(t_0^2) \quad (3.13a)$$

$$t_f^1 = t_0^2. \quad (3.13b)$$

### 3.2.3 Modified Inside-Outside Threat Product Function.

Two threat scenarios are considered: one in which the mission may be accomplished without risk of exposure and another in which risk of exposure is unavoidable due to a fortified intermediate target. Figures 3.1 and 3.2 indicate course layouts, where the small circle indicates the starting position, the ‘x’ indicates the intermediate target, the blue dot indicates final rendezvous, and the shaded regions indicate the threats which must be avoided.

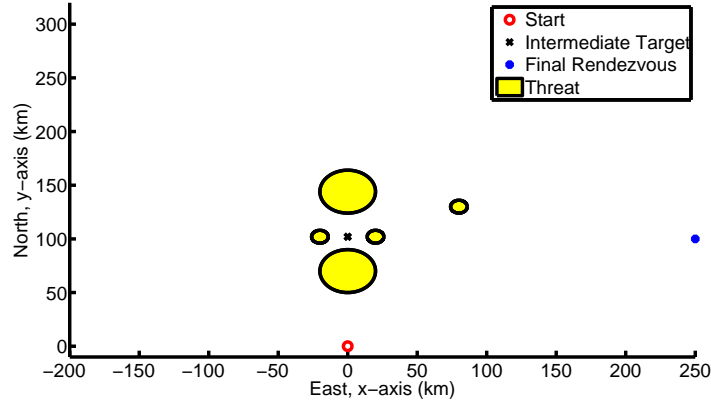
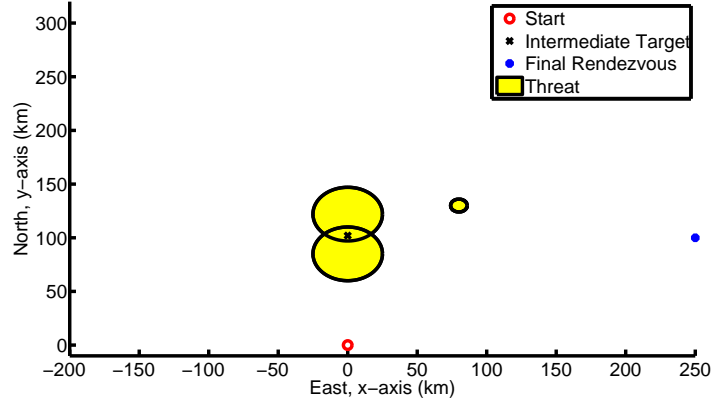


Figure 3.1. Course Layout with Avoidable Threats

Chapter II highlighted superquadrics as a method for modeling threats, which are convenient for the loyal wingman application because of the small number of parameters necessary for representing 2-D and 3-D regions, and thus threat constraint violation can be easily computed using the 2-D version of the inside-outside function,

$$F(\mathbf{h}) = \left( \frac{x_W - x_T}{a_x} \right)^p + \left( \frac{y_W - y_T}{a_y} \right)^2, \quad (3.14)$$



**Figure 3.2. Course Layout with Unavoidable Threats**

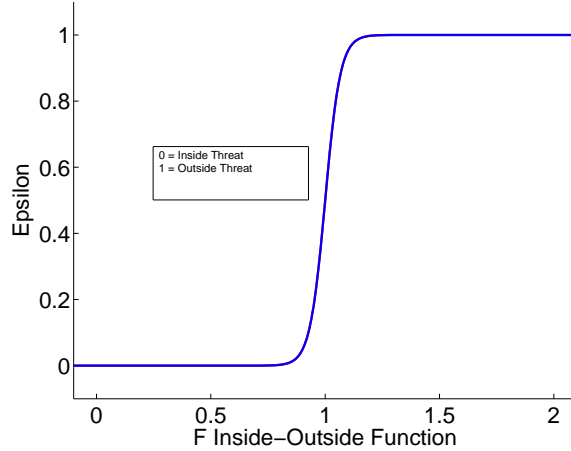
where  $\mathbf{h} = [x_W, y_W, x_T, y_T, a_x, a_y, p]$ ;  $x_W$  and  $y_W$  indicate the position of the loyal wingman along its trajectory;  $x_T$  and  $y_T$  indicate the center point of the threat;  $a_x$  and  $a_y$  are the principal axes of the shape modeling the threat; and  $p$  is a parameter utilized to form the shape of the superquadric, where  $p = 2$  forms a circle or ellipse. A given point,  $(x_W, y_W)$ , on the loyal wingman trajectory is outside the threat when  $F > 1$ , therefore the inequality path constraint is

$$1 - F(\mathbf{h}) \leq 0. \quad (3.15)$$

For scenarios where threats are unavoidable because the target lies inside a threat region, as in Figure 3.2, feasible solutions require that threats not be accounted for as constraints, but rather as part of the cost function. Two challenges arise associated with modeling the inside-outside function as a running cost. First, Equation 3.14 may grow without bound which provides a tendency for the NLP to converge on a trajectory as far away from the threat region as possible, which is not the desire. This challenge can be addressed through a boolean transformation in which all values of  $F$  greater than 1 are set to 1 and all values of  $F$  less than 1 are set to 0. This, however, creates a challenge associated with a gradient-based numerical method for solving the

optimal control problem [83], which requires continuous and differentiable functions. Therefore a modified inside-outside function is developed via a sigmoid function which emulates boolean-like behavior, but provides a continuous and differentiable function between the inside and outside of the threat. Represented through Equation 3.16 and graphically in Figure 3.3, values inside the threat approach zero, values outside the threat approach one and there is a smooth transition from zero to one at the border region of the threat. The stiffness of the transition is represented by  $s$  in Equation 3.16. Its value is a user-defined choice which balances threat exposure modeling error with smooth gradient information required for solving optimal control problems with gradient-based NLP solvers. As an example, referring to Figure 3.3, with  $s = 30$  in an avoidable threat layout, it is likely that the optimizer will converge on trajectories, that when evaluated with Equation 3.14, values of  $F$  will be larger slightly larger than 1.

$$\epsilon(F(\mathbf{h})) = \frac{1}{1 + e^{(s(F(\mathbf{h})-1))}} \quad (3.16)$$



**Figure 3.3. Modified Threat Exposure Function**

This modified inside-outside function may now be used as a running component of the cost formulation to minimize threat exposure. Multiple threats ( $n$ ) are accounted for by taking the product of the modified inside-outside function for each threat [8].

Using the modified inside-outside threat product function, a value of 1 indicates a path in which all threats are avoided. In order to formulate this cost as a minimization, the modified inside-outside threat product function is subtracted from 1 in a running cost, indicated in Equation 3.17,

$$J_{MinExposure} = \int_{t_0}^{t_f} \left(1 - \prod_{i=1}^n \epsilon_i(F_i(\mathbf{h}))\right) dt. \quad (3.17)$$

Establishing the modified inside-outside threat product function as a part of the cost formulation eliminates the need to establish threats as inequality constraints, while simultaneously avoiding threats, and minimizing exposure when threats are unavoidable. Additionally, when using direct orthogonal collocation to transcribe the optimal control problem to a nonlinear programming problem, inequality constraints increase the size and complexity of the Jacobian, generally increasing computation time and decreasing likelihood of generating a feasible solution. Thus, it is recommended that threats in the loyal wingman application are always formulated as part of the cost function and not as inequality constraints.

Threat exposure is the primary component of a fixed-time cost formulation, however a second component, minimize control, is added to ensure a smooth control output:

$$J_{Control} = \int_{t_0}^{t_f} u(t)^2 dt. \quad (3.18)$$

By adding the minimize exposure and minimize control components and applying a convex weighting  $\beta \in [0, 1]$ , the cost function formulation of a fixed time scenario is

$$J_{FixedTime} = (1 - \beta)J_{control} + (\beta)J_{MinExposure}.$$

Next, a formulation is required for scenarios in which minimizing mission time is a priority. A minimum time formulation is developed with a final condition cost

component,  $t_f$ . Because minimizing threat exposure and control output are still desirable, the minimize time scenarios are formulated by adding the final condition component to the  $J_{FixedTime}$  component and applying convex weighting  $\alpha \in [0, 1]$ , resulting in Equation 3.19.

$$J_{MinTime} = \alpha t_f + (1 - \alpha) J_{FixedTime} \quad (3.19)$$

These components of the optimal control problem formulation represents the six scenarios that will be evaluated in the research herein:

1. Fixed time to fixed location with avoidable threats,  $J_{FixedTime}$
2. Fixed time to fixed location with unavoidable threats,  $J_{FixedTime}$
3. Minimize time to fixed location with avoidable threats,  $J_{MinTime}$
4. Minimize time to fixed location with unavoidable threats,  $J_{MinTime}$
5. Minimize time to lead rendezvous with avoidable threats,  $J_{MinTime}$
6. Minimize time to lead rendezvous with unavoidable threats,  $J_{MinTime}$

### 3.3 Loyal Wingman Problem Formulation Summary

The optimal control problem formulation for scenarios 5 and 6 is to minimize the cost

$$J = \alpha t_f + (1 - \alpha) \int_{t_0}^{t_f} \left[ (1 - \beta) u(t)^2 + \beta \left( 1 - \prod_{i=1}^n \epsilon_i(F_i(\mathbf{h})) \right) \right] dt \quad (3.20)$$

where  $\epsilon$  and  $F$  are defined in Equations 3.16 and Equation 3.14, respectively, subject to dynamic constraints,

$$\dot{x} = V \cos \psi \quad (3.21a)$$

$$\dot{y} = V \sin \psi \quad (3.21b)$$

$$\dot{\psi} = u, \quad (3.21c)$$

boundary conditions,

$$\mathbf{x}_W^1(t_0) = \mathbf{x}_0 \quad (3.22a)$$

$$x_W^g(t_f^g) = x_g \quad (3.22b)$$

$$y_W^g(t_f^g) = y_g \quad (3.22c)$$

$$\mathbf{x}_W^G(t_f^G) = \mathbf{x}_L(t_f^G), \quad (3.22d)$$

and linkage constraints,

$$\mathbf{x}_W^g(t_f^g) = \mathbf{x}_W^{g+1}(t_0^{g+1}) \quad (3.23a)$$

$$t_f^g = t_0^{g+1}, \quad (3.23b)$$

$\forall g = 1, 2, \dots, G-1$ ,  $G$  representing the number of phases. When the final state location is fixed, such as in scenarios 1, 2, 3, and 4, Equation 3.22d must be updated,

$$\mathbf{x}_W^G(t_f^G) = \mathbf{x}_G. \quad (3.24)$$

Finally, if the final time is fixed, as in scenarios 1 and 2, the cost function to minimize is

$$J = \int_{t_0}^{t_f} [(1 - \beta)u(t)^2 + \beta(1 - \prod_{i=1}^n \epsilon_i(F_i(\mathbf{h})))] dt \quad (3.25)$$

and an additional final boundary condition constraint is enforced,

$$t_f^G = t_G. \quad (3.26)$$

### 3.4 Initial Guess Results

The work herein used the Gauss Pseudospectral Optimization Software (GPOPS II)[143] which is a multi-purpose *MATLAB*<sup>®</sup>-based [79] transcription software. GPOPS II uses the roots of the Legendre polynomial as discretization points as well as choosing the Radau Pseudospectral Method (RPM) which places collocation nodes at the initial condition and in the interior, but not at the final condition. The intermediate-target optimal control problem includes an intermediate target, which is modeled using multiple phases such that the final conditions for phase 1 are the initial conditions for phase 2 and so on between each phase. A challenge with gradient-based NLPs, especially with nonlinear, nonconvex problems is that an initial guess is necessary and the choice for the initial guess impacts computation time as well as which locally optimal solution the NLP converges upon. In order to demonstrate this challenge, a single scenario, fixed time to fixed location through unavoidable threats, is formulated and solved by supplying various initial guesses to the NLP. Nine ‘simple’ initial guesses are developed by a method that can be visualized in Figure 3.4. In phase one, three paths are created that connect the initial condition to the intermediate target: an arc to the left, an arc to the right, and a straight path, labeled 1 through 3. The same is done for phase two and each segment labeled 4 through 6. This represents 9 potential initial guesses. Each of these initial guesses is supplied to the NLP and the resulting cost and computation time for each initial guess is documented in Table 3.1 and plotted in Figure 3.5. In this particular scenario, each initial guess supplied to the NLP converged to a different locally optimal solution.

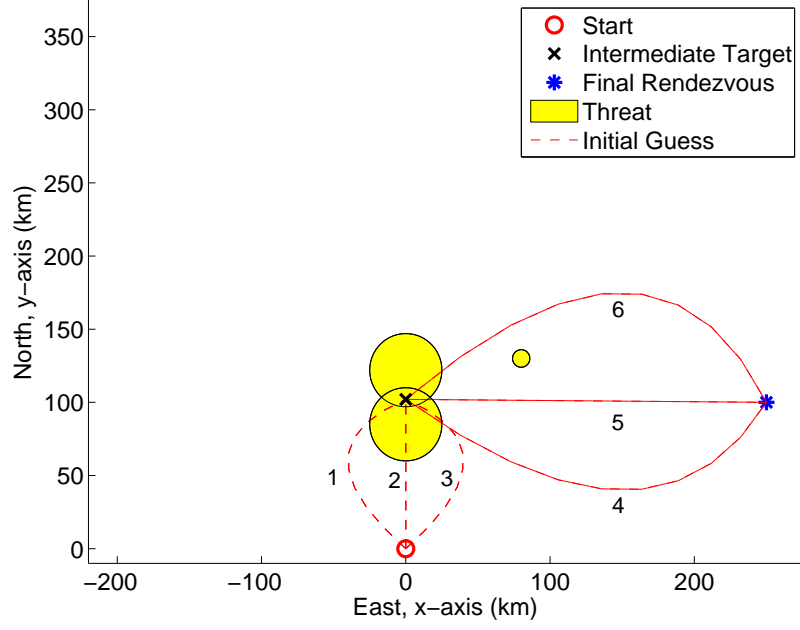
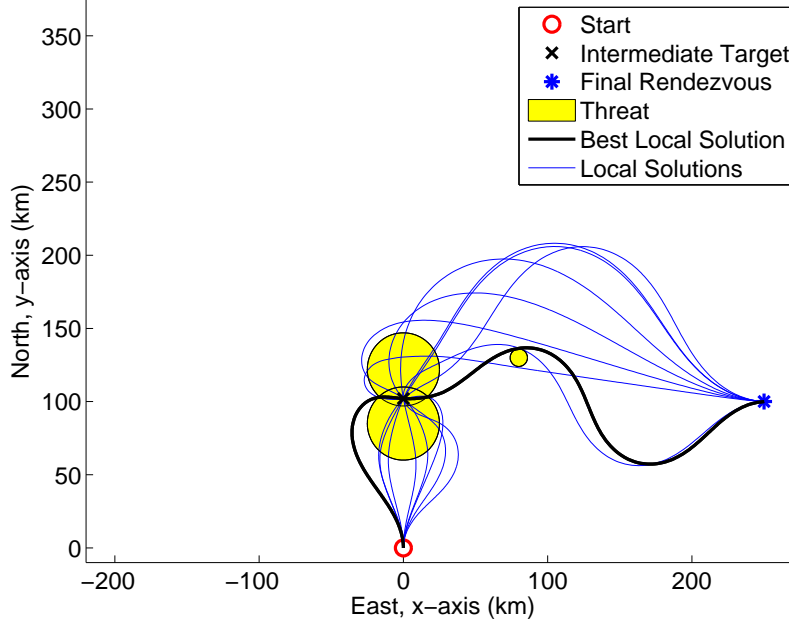


Figure 3.4. Initial Guess Segments Supplied to the NLP

Table 3.1. Cost and Computation Time of 9 Initial Guesses

Segment	Cost	Computation Time
1-4	16.49	1.76
1-5	14.94	1.99
1-6	11.19	2.46
2-4	15.03	2.36
2-5	14.17	2.24
2-6	15.51	4.26
3-4	20.33	2.12
3-5	15.65	4.38
3-6	23.83	3.64



**Figure 3.5. Nine Locally Optimal Solutions for Unavoidable Threats**

The best cost from these nine results was from the initial guess segment 1-6, plotted in black in Figure 3.5. NLP optimization tolerance, number and location of nodes, the convex weighting, as well as the initial guess are all factors that contribute to which locally optimal solution the NLP ultimately converges upon. Additional work may be applied to determining the right combination of NLP optimization tolerance, node placement, convex weighting and initial guess. This work, however, chooses to ameliorate this challenge using a hybrid optimization methodology for achieving rapid and autonomous solutions, which is demonstrated in the next chapter.

### 3.5 Optimal Control Problem Formulation Conclusions

Research question one was answered by formulating the optimal control problem in a static threat environment using a 3-state, 2-D model. The intermediate target conditions were introduced by breaking the problem into multiple phases and linking initial and final conditions for each phase. Threat regions were modeled using

superquadrics and the inside-outside function. To ensure threat avoidance and when necessary minimize exposure, a modified inside-outside threat product function was developed and employed as a component of the running cost. Two cost functions were established, one for a fixed time scenario and the other for a minimize time scenario. This chapter utilized a single scenario to highlight challenges with a highly nonlinear, nonconvex optimal control problems. Multiple initial guesses were provided to the NLP, resulting in convergence to multiple locally optimal solutions. This challenge will be ameliorated through the use of a hybrid optimization technique demonstrated in the next chapter.

## IV. Hybrid Optimization Methodology

The purpose of this chapter is to demonstrate the hybrid optimization methodology in which a heuristic-based Particle Swarm Optimization (PSO) algorithm is used to supply an initial guess to the gradient-based nonlinear programming (NLP) solver. The previous chapter formulated the optimal control problem which was then transcribed to an NLP problem using direct orthogonal collocation (DOC). This chapter exploits the myriad of variations of the PSO method to produce a PSO algorithm tailored to the loyal wingman optimal control problem with the specific task of rapidly providing an initial guess to the NLP. The hybrid technique as applied to the loyal wingman optimal control problem is considered successful because it produces a rapid and feasible, converged NLP solution. The chapter begins by describing a PSO algorithm written for the loyal wingman optimal control problem, which highlights methods for producing seeds, handling constraints, and calculating costs. Then, a simulation is run and a metric is established to compare the speed and accuracy of the hybrid technique to using DOC alone and concludes that the loyal wingman optimal control problem may be solved effectively using the hybrid optimal control methodology.

### 4.1 The Loyal Wingman PSO Algorithm

A flowchart of the loyal wingman PSO algorithm is shown in Figure 4.1, which is broken into two major sections: algorithm initialization and algorithm iterations. There are many factors in considering how a PSO algorithm should be varied for a particular application. Section 4.2 will show the PSO algorithm written for the loyal wingman application as effective for achieving rapid, accurate results.

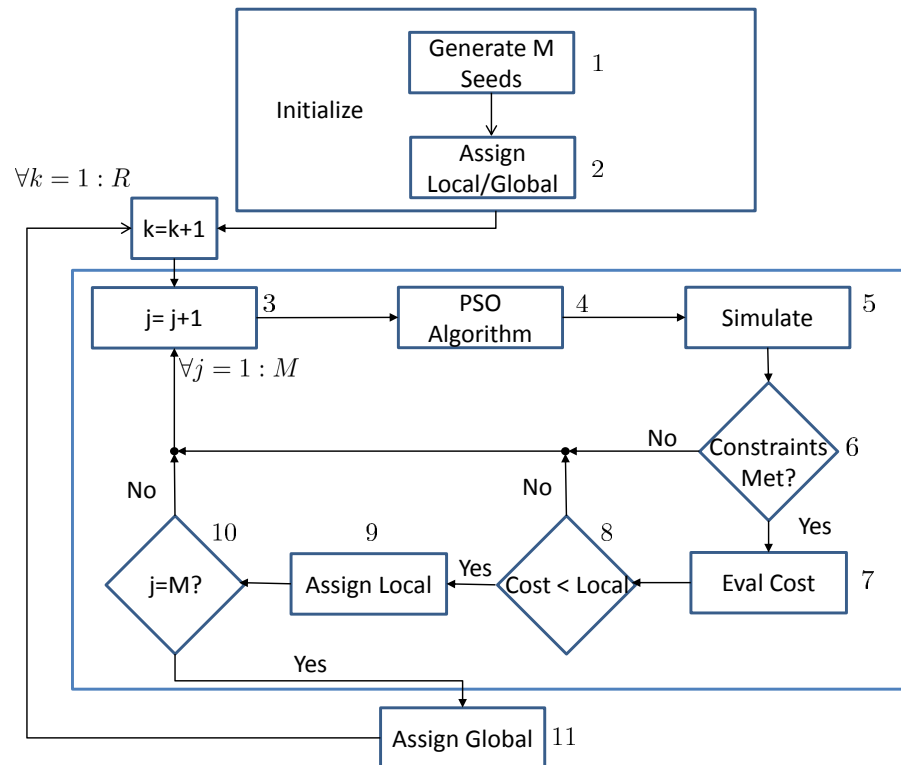


Figure 4.1. PSO Algorithm Flowchart

#### 4.1.1 PSO Seeds and Initialization.

Referring to Figure 4.1, the first section is the production of seeds and initialization of the components of the basic two-line PSO algorithm used in the loyal wingman PSO algorithm,

$$\vec{v}(k+1)_j = K[\vec{v}(k)_j + b_L \vec{r}_1 \otimes (\vec{u}(k)_j - \vec{L}_j) + b_G \vec{r}_2 \otimes (\vec{u}(k)_j - \vec{G})] \quad (4.1a)$$

$$\vec{u}(k+1)_j = \vec{u}(k)_j + \vec{v}(k+1)_j, \quad (4.1b)$$

$\forall j = 1, 2, \dots, M$  particles,  $\forall k = 1, 2, \dots, R$  iterations, where  $\otimes$  represents element-wise multiplication of vector components and  $\vec{r}_1, \vec{r}_2$  represent vectors of nondeterministic evenly distributed parameter weighting in  $[0, 1]$ . The remaining components,  $\vec{v}$ ,  $\vec{u}$ ,  $\vec{L}$ ,  $\vec{G}$ ,  $b_L$ ,  $b_G$ , and  $K$ , are described next.

There are various ways to define a particle and a recommendation of this research is to explore other methods for increased efficiency. However, the work herein defines a particle as a vector of discrete control inputs,  $\vec{u} = [u_1, u_2, \dots, u_N]^T$ , where  $N$  is the number of discrete control inputs. Simulation of Equation 3.21 with the heading rate control vector,  $\vec{u}$ , produces a trajectory for use in evaluating the cost and constraints. A deterministic method, described in Appendix A, was developed for producing particle seeds based on the following criteria: satisfy target and endpoint constraints as recommended by Hu and Ebert [109], create a broad range of possible trajectories to aid in the PSO stochastic search, and produce in a computationally efficient manner. Using this deterministic method, an initial set of  $M$  control vectors is produced  $\vec{u}(0)_{1 \times M} = [\vec{u}(0)_1, \vec{u}(0)_2, \dots, \vec{u}(0)_M]$ , which when simulated produce an initial set of  $M$  discrete vectors for each of the 3-states,  $\{\vec{x}(0)_{1 \times M}, \vec{y}(0)_{1 \times M}, \vec{\psi}(0)_{1 \times M}\}$

The PSO algorithm is initialized by assigning each initially produced seed as its own current local best,  $\vec{L}_j = \vec{u}(0)_j$ ,  $\forall j = 1, 2, \dots, M$ . The cost associated with each

seed, as described later in this chapter, is determined and the index of the particle with the best cost is assigned  $j^*$ , such that the global best is  $\vec{G} = \vec{L}_{j^*}$ .

The first component of Equation 4.1 is the constriction factor

$$K = \frac{2}{\phi - \sqrt{\phi^2 - 4\phi}} \quad (4.2)$$

with  $\phi = b_L + b_G$ ,  $\phi > 4$ , suggested by Clerc [100], where  $b_L$  and  $b_G$  represent the local (L) and global (G) weighting factors. The choice of these parameters effect the local and global nature of the search as well as convergence tendencies [99]. These values were chosen for use in the loyal wingman PSO algorithm through an experiment as described in Appendix C.

Finally, the velocity component  $\vec{v}(0)_j$  is initialized to  $\vec{0}$ ,  $\forall j = 1, 2, \dots, M$ . There are many ways to initialize a PSO algorithm and additional work could be done to tune the various parameters for a given set of scenarios. This work does not claim to have found the perfect combination for optimal performance, however, the initialization described in this section is sufficient to accomplish the desired task, which is to rapidly provide an initial guess to the gradient-based NLP that results in an accurate and feasible solution. Additional work in determining the right seeds and parameters may further improve the quality and efficiency of the results.

#### 4.1.2 PSO Iterations.

After the PSO is initialized, a series of steps occur which allows the ‘flock’ of control vectors, or ‘particles’, to change values through a stochastic search of the space, moving toward the currently assigned global best solution. Beginning with the first iteration,  $k = 1$ , a single particle,  $\vec{u}_j$ , is updated through Equation 4.1. The

updated particle is simulated using the discrete equations of motion

$$x_j(i+1) = V \cos(\psi_j(i))\Delta t + x_j(i) \quad (4.3a)$$

$$y_j(i+1) = V \sin(\psi_j(i))\Delta t + y_j(i) \quad (4.3b)$$

$$\psi_j(i+1) = u(i)\Delta t + \psi_j(i) \quad (4.3c)$$

$\forall i = 1, 2, \dots, N-1$ , where  $V$  is the velocity, held constant, and  $\Delta t$  remains fixed for all particle and iterations.

The updated particle is then evaluated in Figure 4.1, Box 6, to check constraint criteria. Intermediate and final conditions constraint criteria are relaxed in the PSO to improve the PSO search capability for a reasonable computation time. The loss of fidelity through this relaxation is overcome by speed and accuracy of the direct orthogonal collocation methodology. The Euclidean distance from all points in the trajectory to the intermediate target  $(x_1, y_1)$  are calculated

$$D_j(i) = \sqrt{[x_j(i) - x_1]^2 + [y_j(i) - y_1]^2} \quad (4.4)$$

$\forall i = 1, 2, \dots, N$ . If the minimum  $D(i)$ ,  $\forall D(i), 1, 2, \dots, N$  is less than the  $D^*$  threshold for meeting the intermediate target constraint, then the same formulation is used to check the final condition constraint,  $(x_{fc}, y_{fc})$ . If either constraint threshold is not met, the PSO velocity component is set to 0,  $\vec{v}_j = \vec{0}$  [105] and the next particle is evaluated  $\forall j = 1, 2, \dots, M$ .  $D^*$  is set to 5 km, much higher than the criteria for the primary problem formulation. This is done to increase the number of particles on each iteration satisfying boundary conditions and not being ‘thrown out’, improving computation time and subsequently relying on the speed and accuracy of the DOC method to regain lost fidelity.

When a particle meets both intermediate and final condition constraints, the cost

is then evaluated (Figure 4.1, Box 7). A separate cost function is established for each boundary condition scenario:

1. Minimize time to fixed location, Equation 4.5
2. Fixed time to fixed location, Equation 4.6
3. Minimize time to lead rendezvous, Equation 4.7

The minimize time to fixed location, scenario 1, is

$$J_{MinTimePSO} = \alpha t_f + (1 - \alpha) \int_{t_0}^{t_f} [(1 - \beta)u(t)^2 + \beta(1 - \prod_{i=1}^n \epsilon_i(F_i(\mathbf{h})))] dt. \quad (4.5)$$

A fixed time,  $t_{fc}$ , scenario contains an additional constraint. Instead of checking to ensure the constraint is met on each iteration (Box 6), the fixed time constraint is formulated as an additional component in the cost function [110],  $J_{\Delta t} = t_f - t_{fc}$ , such that,

$$J_{FixedTimePSO} = \alpha_{FT} J_{\Delta t} + (1 - \alpha_{FT}) \int_{t_0}^{t_f} [(1 - \beta)u(t)^2 + \beta(1 - \prod_{i=1}^n \epsilon_i(F_i(\mathbf{h})))] dt. \quad (4.6)$$

The convex weighting  $\alpha_{FT} \in [0, 1]$  is adjusted to put increased emphasis on the fixed final time, such that as the particle's variation from the desired final time increases, the cost increases and when there is no variation from the desired final time, this component goes to zero and only the running cost remains.

The scenario, minimize time to rendezvous with the lead,  $J_{RendezvousPSO}$ , has an additional constraint that the states,  $\mathbf{x}$ , of the loyal wingman (subscript  $W$ ) and lead (subscript  $L$ ) match at the final time,  $\mathbf{x}_W(t_f) = \mathbf{x}_L(t_f)$ . The cost function to minimize time to rendezvous while minimizing threat exposure is

$$J_{RendezvousPSO} = (1 - \alpha_{Rend}) J_{MinTimePSO} + \alpha_{Rend} D_{Rend}, \quad (4.7)$$

where  $D_{Rend}$  is the minimum Euclidean distance between the loyal wingman and lead trajectory at each time-step, from the index of the intermediate waypoint,  $i^*$ , forward,

$$D_{Rend} = \min\{\sqrt{[x_W(i) - x_L(i)]^2 + [y_W(i) - y_L(i)]^2}\} \quad (4.8)$$

$\forall i = i^*, i^* + 1, \dots, N$  and  $\alpha_{Rend} \in [0, 1]$  is weighted such that as  $D_{Rend}$  increases, the cost increases and when  $D_{Rend}$  goes to 0, this component goes to zero.

The cost of the current particle is then evaluated according to the appropriate cost for the scenario using Gaussian quadrature and then compared to the cost of the particle's current local best,  $\vec{L}_j$  (Figure 4.1, Box 8). If the updated particle's cost is lower, then the particle's local best is updated,  $\vec{L}_j = \vec{u}_j$  (Figure 4.1, Box 9). The process for a single iteration,  $k$ , is repeated for all particles  $j = 1, 2, \dots, M$ , which is: update particle through Equation 4.1 (Box 4), simulate using Equation 4.3 (Box 5), check constraint criteria (Box 6), evaluate cost (Box 7), and update local best (Boxes 8 and 9). After all particles have been updated and evaluated (Box 10), if any local bests have been reassigned, then the costs of the new local bests are compared to the current global best. If a particle's local best cost is lower, the global best is updated,  $\vec{G} = \vec{L}_{j^*}$  (Box 11). This completes one iteration of the PSO algorithm. The iterations continue  $\forall k = 1, 2, \dots, R$  until the iteration limit has been reached. The global best,  $\vec{G}$  at the final iteration is the solution used as the initial guess to supply to the gradient-based NLP.

## 4.2 Results

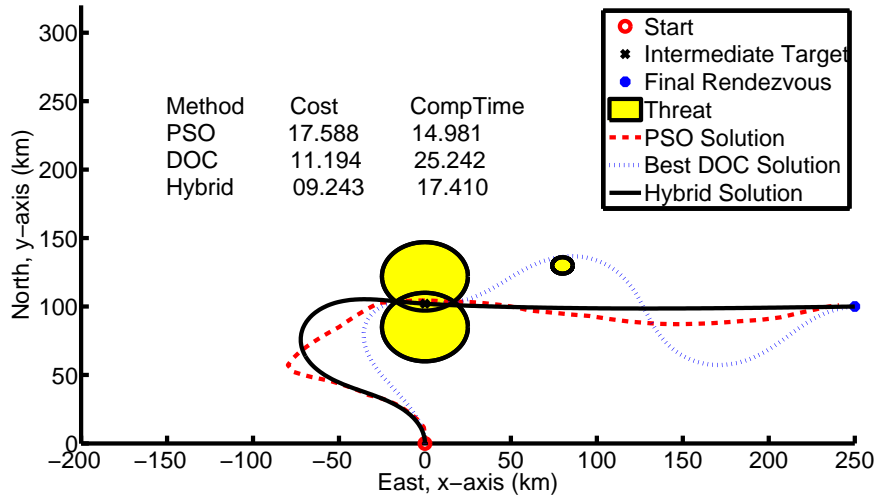
The loyal wingman optimal control problem is solved using three methods, direct orthogonal collocation (DOC) using a gradient-based NLP, a heuristic particle swarm optimization (PSO) and a hybrid technique in which the output from the PSO algo-

rithm is used as an initial guess for DOC. The DOC method was run nine times using nine different initial guesses, and repeated for each scenario. Using the GPOPS II software, anywhere from 4 to 9 different locally optimal feasible solutions were produced by the NLP solver. The costs of these different solutions were compared and the lowest cost output was identified as the ‘best’ solution from the DOC method. Computation time for the DOC method is measured as the total time it takes to run GPOPS II and the NLP solver for all nine initial guesses. An example scenario, fixed time to fixed point through unavoidable threats, was discussed in Chapter III and the results captured in Table 3.1 and Figures 3.4 and 3.5.

Next, the PSO algorithm was run for each scenario for a pre-determined 100 iterations. The cost and computation time were captured along with a graphical representation of the currently assigned global best solution when the 100 max iteration limit was achieved.

Finally, a hybrid technique is tested by taking the output of the previously mentioned PSO algorithm and supplying it as the initial guess into the DOC’s gradient-based NLP. The cost output is captured and the computation time is computed as the combined time to run the PSO, and the DOC with the PSO output as the initial guess. The following six figures are provided which overlays the trajectory results from the DOC method alone (dotted line), the PSO method alone (dashed line) and the hybrid method (solid line). A table in each figure identifies the cost and computation time associated with each method.

In order to achieve an *accurate* solution, it is expected that when threat regions are avoidable, each method should find trajectories that successfully avoids the threat regions. In cases where threat regions are unavoidable, it should be expected that the best way to minimize time of exposure for a constant velocity vehicle and equally weighted threat regions is to traverse the threats by way of a perpendicular bisector

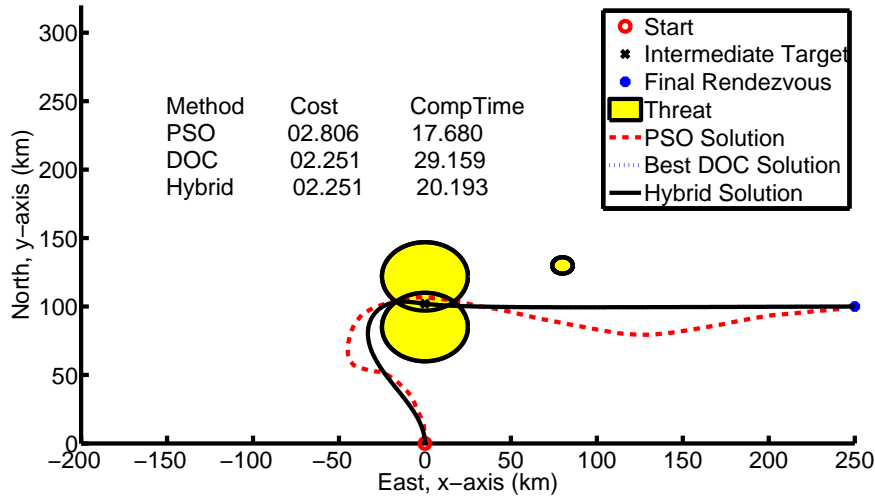


**Figure 4.2. Comparison of Fixed Time Through Unavoidable Threats**

of the threat intersection points. Minimum time scenarios should result in trajectories which are direct, while fixed final time scenarios should produce additional turns that allow for idle time in order to meet the fixed final time constraint.

Figure 4.2 represents scenario one in which the vehicle must overfly an intermediate target in a layout in which threats are not avoidable, and conclude with rendezvous at the fixed final time and specified location. All three methods find trajectories that minimize exposure through unavoidable threat regions with a perpendicular bisector and the DOC method produces a trajectory with a lower cost than does the PSO alone. However, the hybrid method produces the lowest cost solution at a computation time that is faster than the DOC method alone.

Figure 4.3 represents the results of scenario two in which the vehicle must overfly an intermediate target in a layout in which threat regions are unavoidable and must conclude at the final rendezvous in minimum time. The DOC solution produces a lower cost solution than the PSO method but takes nearly twice the computation time. The hybrid method produces the exact same solution as the DOC method alone, but does so in a more computationally efficient manner. The time it takes to



**Figure 4.3. Comparison of Minimize Time Through Unavoidable Threats**

run the hybrid method is nearly 30% faster than the DOC method alone.

Figure 4.4 represents the results of scenario three in which the vehicle must overfly an intermediate target prior to a fixed final time and fixed final location rendezvous. This mission can be accomplished without exposure to any threat regions and all methods do so successfully. The results are similar to the scenario previously discussed where the cost of the DOC solution is lower than the cost of the PSO solution, but to run the DOC solution nine times, once again takes twice the computation time. In this case, the hybrid method outputs the same trajectory and cost associated with the DOC method alone, but does so 12 seconds faster, in approximately 40% of the time it takes to run the DOC method alone.

Figure 4.5 is scenario four, representing a vehicle that must overfly an intermediate target prior to rendezvous at a final location. The mission may be accomplished without exposure to threats, which all methods obtain successfully. What is interesting about this scenario is the DOC method by itself, which was run with nine different initial guesses, returns a best-cost trajectory that flies North (positive y-direction) of the isolated threat on its way to the final rendezvous. This trajectory adds unnec-

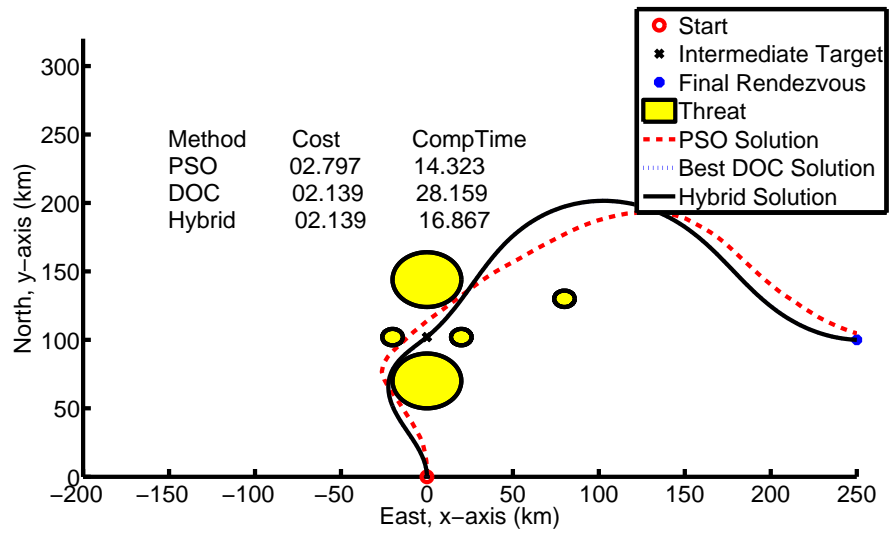


Figure 4.4. Comparison of Fixed Time Through Avoidable Threats

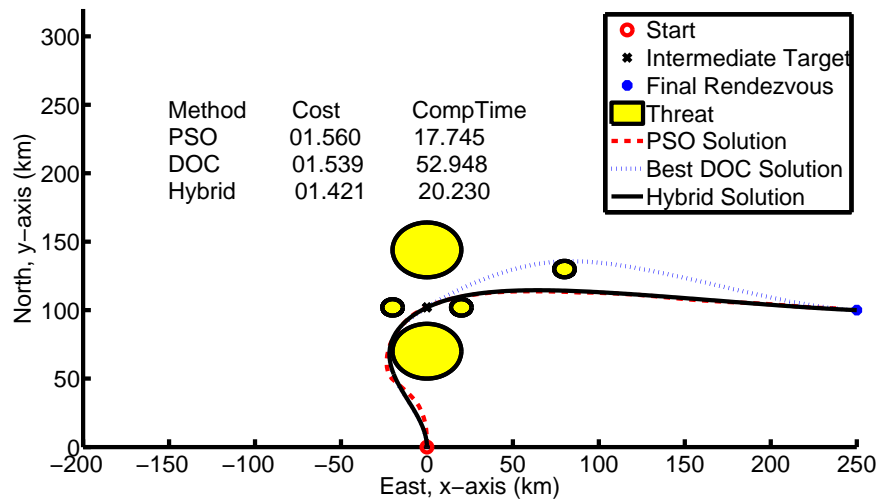
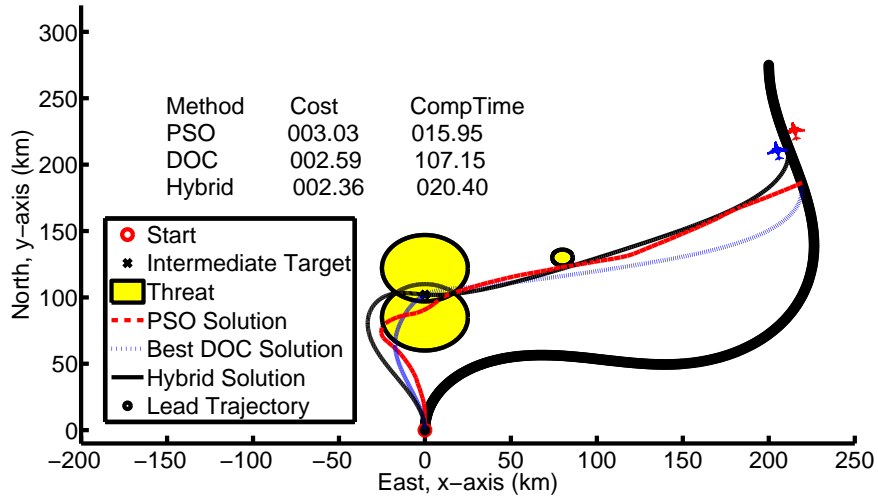


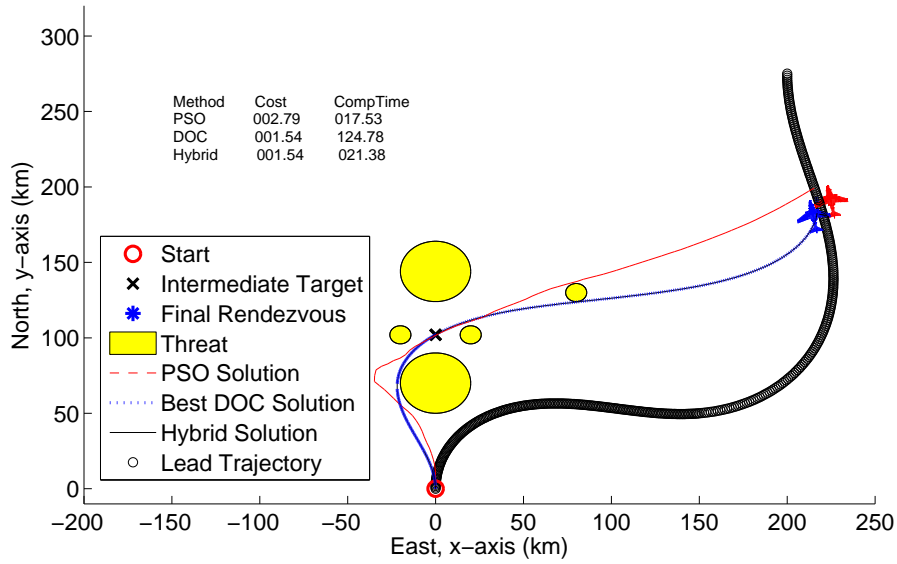
Figure 4.5. Comparison of Minimize Time Through Avoidable Threats



**Figure 4.6. Comparison of Minimize Time to Rendezvous Through Unavoidable Threats**

essary cost in a minimum time problem and its result is an example of one of the challenges faced by the gradient-based search methods. As the NLP takes small steps in a direction to minimize the time, the trajectory is exposed to the threat region and bounces back above, effectively being caught in a local minimum. The global and stochastic nature of the PSO results in a more direct, time-saving route. When this PSO result is supplied as the initial guess into the NLP in a hybrid approach, a lower cost solution is found in less than half the computation time to run the DOC method alone.

Figures 4.6 and 4.7 are the results for scenarios five and six which both entail a minimum time to rendezvous with the manned lead. The threat layout is the same as was provided in previous figures with the addition of the manned lead's trajectory. In Figure 4.6, the trajectory must minimize exposure through unavoidable threat regions. In this scenario, neither the PSO nor DOC method alone produce a desired trajectory passing through the threat perpendicular bisector. The PSO solution, however, is sufficient to supply as an initial guess to the NLP, demonstrating the desired behavior of the hybrid methodology. Additionally, each of the DOC runs alone



**Figure 4.7. Comparison of Minimize Time to Rendezvous Through Avoidable Threats**

took an average of 12 seconds, totaling almost 110 seconds when running nine times and the end result is a trajectory that does not find the perpendicular bisector. When using the hybrid method, the PSO took about 16 seconds, but the NLP only took 4 additional seconds to converge on a lower cost solution that minimizes exposure. The hybrid method took less than 20% of the time to run as the DOC method and produced a lower cost solution.

Scenario six results are shown in Figure 4.7, where all methods return trajectories that successfully avoids threat regions. When running the DOC method alone nine times, a good solution is found at the expense of 124 seconds in computation time. The hybrid method found the same cost solution in 21 seconds - 17% of the time it takes to run the DOC method alone.

Table 4.1 provides a comparison of cost and computation time for all three methods for each of the six scenarios and these results are summarized in the next section.

**Table 4.1. Cost and Computation Time Comparison for All Scenarios**

	Scen 1		Scen 2		Scen 3	
	Cost	Time	Cost	Time	Cost	Time
PSO	17.588	14.981	2.806	17.680	2.797	14.323
DOC	11.194	25.242	2.251	29.159	2.139	28.159
Hybrid	9.243	17.410	2.251	20.193	2.139	16.867
	Scen 4		Scen 5		Scen 6	
	Cost	Time	Cost	Time	Cost	Time
PSO	1.560	17.745	3.030	15.950	2.790	17.530
DOC	1.539	52.948	2.590	107.150	1.540	124.780
Hybrid	1.421	20.320	2.360	20.400	1.540	21.380

### 4.3 Hybrid Methodology Conclusions and Recommendations

A PSO algorithm was developed for use in the loyal wingman application with the goal of rapidly providing a feasible solution to supply to the gradient-based NLP. PSO seeds were developed deterministically in order to decrease PSO computation time and methods of handling constraints recommended by other authors were applied. A simulation was run and a metric established for comparing DOC alone to the hybrid technique, which used the PSO result as its initial guess. In general, all three methods provided feasible results, but the hybrid method was superior. The DOC solution was dependent upon the initial guess as was suggested, producing multiple locally optimal solutions. One run of the DOC method is fast, but if the user doesn't have a good initial guess and therefore needs to run the method multiple times to find a good initial guess, then the computation time is high. The PSO algorithm used for this application was rapid and found the best cost solution within the space that was searched. The hybrid method consistently resulted in a lower cost than the PSO alone and either was even with or lower than the results from the DOC method alone. In all cases, the hybrid method produced results faster than the DOC method alone. Although the hybrid method consistently provided rapid, accurate solutions,

the method remains imperfect. There is no way to guarantee a full global search and even though the NLP tends to converge on the locally optimal solution in the region of the initial guess, NLP step-size may cause the optimizer to ‘jump’ out of the region of the initial guess.

There are many factors to consider when tailoring the PSO to a particular application such as, definition of particles, production of particle seeds, the use of a constriction factor and its value, local and global parameter weighting values, and the various ways of handling constraints as utilized in this work. These factors were tailored for the loyal wingman application and there are potentially more factors to consider. A recommendation for future work should consider continuing to adjust the PSO to increase the stochastic search region as well as decreasing the computation time. A second recommendation for future work is to study the definition of a particle in this application as the coefficients to a polynomial, discretized at the roots of an orthogonal basis set. There is potential for synergy by using the DOC transcription of the optimal control problem and applying it for use in the PSO.

Overall, this chapter has demonstrated a methodology for solving the loyal wingman optimal control problem, and between Chapters III and IV, answered research question one by formulating and solving the optimal control problem in a static threat environment.

## V. 3-D Results

The purpose of this chapter is to add real-world fidelity to the results of the previous chapter by answering question one using a 3-D model. First, the optimal control problem that was formulated in Chapter III is updated and summarized for a 3-D model, then there is a discussion on the required updates to the PSO algorithm. Results are provided along with a discussion for the six identified scenarios, followed by a conclusion and recommendations for future work.

### 5.1 Update Optimal Control Problem Formulation

A 7-state, 3-DOF model suitable for the loyal wingman problem was provided in Chapter II and will be used to provide 3-D results. The initial and final conditions for scenarios identified in Chapter III remain unchanged, however the intermediate target constraint is updated to include the altitude state,  $z$ , such that

$$\begin{aligned}x_W^1(t_f) &= x_W^2(t_0) &= x_1 \\y_W^1(t_f) &= y_W^2(t_0) &= y_1 \\z_W^1(t_f) &= z_W^2(t_0) &= z_1,\end{aligned}\tag{5.1}$$

where the superscripts indicate phases. Finally, the 2-D inside-outside function, Equation 3.14, is updated for a 3-D threat,

$$F(\mathbf{h}) = \left(\frac{x_W - x_T}{a_x}\right)^p + \left(\frac{y_W - y_T}{a_y}\right)^2 + \left(\frac{z_W - z_T}{a_z}\right)^2,\tag{5.2}$$

where  $\mathbf{h} = [x_W, y_W, z_W, x_T, y_T, z_T, a_x, a_y, a_z, p]$ ;  $x_W$ ,  $y_W$ , and  $z_W$  indicate the position of the loyal wingman along its trajectory;  $x_T$ ,  $y_T$ , and  $z_T$  indicate the center point of the threat;  $a_x$ ,  $a_y$ , and  $a_z$  are the principal axes of the shape modeling the threat; and

$p$  is a parameter utilized to form the shape of the superquadric, where  $p = 2$  forms a circle or ellipse. The same six scenarios identified in Chapter III are applicable for the 3-D model

1. Fixed time to fixed location with avoidable threats,  $J_{FixedTime}$
2. Fixed time to fixed location with unavoidable threats,  $J_{FixedTime}$
3. Minimize time to fixed location with avoidable threats,  $J_{MinTime}$
4. Minimize time to fixed location with unavoidable threats,  $J_{MinTime}$
5. Minimize time to lead rendezvous with avoidable threats,  $J_{MinTime}$
6. Minimize time to lead rendezvous with unavoidable threats,  $J_{MinTime}$

## 5.2 3D Loyal Wingman Problem Formulation Summary

The optimal control problem formulation for scenarios 5 and 6 is to minimize the cost,

$$J = \alpha t_f + (1 - \alpha) \int_{t_0}^{t_f} [(1 - \beta)[u_1(t)^2 + u_2(t)^2] + \beta(1 - \prod_{i=1}^n \epsilon_i(F_i(\mathbf{h})))] dt, \quad (5.3)$$

where,

$$\epsilon(F(\mathbf{h})) = \frac{\epsilon_{max}}{1 + e^{-s(F(\mathbf{h})-1)}}, \quad (5.4)$$

$s$  is a user defined stiffness balancing threat border region accuracy with computational efficiency, and  $F$  is identified in Equation 5.2. The cost is subject to dynamic

constraints,

$$\dot{x}(t) = V \cos \gamma(t) \cos \chi(t) \quad (5.5a)$$

$$\dot{y}(t) = V \cos \gamma(t) \sin \chi(t) \quad (5.5b)$$

$$\dot{z}(t) = V \sin \gamma(t) \quad (5.5c)$$

$$\dot{\gamma}(t) = \frac{N_z g \cos \mu(t) - g \cos \gamma(t)}{V} \quad (5.5d)$$

$$\dot{\chi}(t) = \frac{N_z g \sin \mu(t)}{V \cos \gamma(t)} \quad (5.5e)$$

$$\dot{N}_z(t) = u_1(t) \quad (5.5f)$$

$$\dot{\mu}(t) = u_2(t), \quad (5.5g)$$

$$(5.5h)$$

boundary conditions

$$\mathbf{x}_W^1(t_0) = \mathbf{x}_0 \quad (5.6a)$$

$$x_W^g(t_f^g) = x_g \quad (5.6b)$$

$$y_W^g(t_f^g) = y_g \quad (5.6c)$$

$$z_W^g(t_f^g) = z_g \quad (5.6d)$$

$$\mathbf{x}_W^G(t_f^G) = \mathbf{x}_L(t_f^G), \quad (5.6e)$$

and linkage constraints,

$$\mathbf{x}_W^g(t_f^g) = \mathbf{x}_W^{g+1}(t_0^{g+1}) \quad (5.7a)$$

$$t_f^g = t_0^{g+1}, \quad (5.7b)$$

$\forall g = 1, 2, \dots, G - 1$ , where  $G$  represents the number of phases and  $\mathbf{x}$  now represents the 7 states  $\{x, y, z, \gamma, \chi, N_z, \mu\}$ . When the final state location is fixed, such as in

scenarios 1, 2, 3, and 4, Equation 5.7a is replaced with

$$\mathbf{x}_W^G(t_f^G) = \mathbf{x}_G \quad (5.8)$$

Finally, if the final time is fixed, as in scenarios 1 and 2, the cost function to minimize is

$$J = \int_{t_0}^{t_f} [(1 - \beta)[u_1(t)^2 + u_2(t)^2] + \beta(1 - \prod_{i=1}^n \epsilon_i(F_i(\mathbf{h})))] dt \quad (5.9)$$

and an additional final boundary condition constraint is enforced,

$$t_f^G = t_G. \quad (5.10)$$

### 5.3 Update Loyal Wingman PSO Algorithm

There are two updates to the loyal wingman PSO algorithm. One is the use of a reduced-order dynamics model and the other is the method of producing PSO seeds.

#### 5.3.1 The Reduced Order PSO Dynamic Model.

A positive characteristic of a full-state model is the accuracy of results, which when provided to the NLP as an initial guess, improves NLP computation time. Use of the full-state model in the loyal wingman PSO was more challenging, however, because it has a significant negative impact on the speed and convergence of the PSO. A recommendation of the work herein is to continue to improve the PSO, including choice of the PSO parameters; however, this work chose to improve the convergence and computation time of the PSO by using a reduced-order dynamics model. The sacrifice of accuracy from a reduced-order PSO is deemed acceptable due to the increased computational efficiency of the reduced-order PSO as well as reliance upon the accuracy achieved by use of the DOC method.

The reduced-order PSO is a 3-state, 2-control model, taken from Equations 5.5a, 5.5b, and 5.5c

$$\dot{x}(t) = V \cos \gamma(t) \cos \chi(t) \quad (5.11a)$$

$$\dot{y}(t) = V \cos \gamma(t) \sin \chi(t) \quad (5.11b)$$

$$\dot{z}(t) = V \sin \gamma(t), \quad (5.11c)$$

where  $x, y, z$  represent position coordinates, and the controls are heading,  $\chi$ , and flight path angle,  $\gamma$ .

### 5.3.2 PSO Seeds in Three Dimensions.

Chapter IV established a deterministic method of producing seeds to initiate the PSO algorithm. This prescriptive method, described in Appendix A, is useful for all loyal wingman scenarios which included identifying points and spline-fitting trajectories through those points. Identification of those points must be updated for use in three dimensions and is described in Appendix B.

### Loyal Wingman PSO Algorithm in 3-D.

Each particle on each iteration is simulated using the discrete version of Equation 5.11a, 5.11b, and 5.11c,

$$x(i+1) = V \cos \gamma(i) \cos \chi(i) \Delta t + x(i) \quad (5.12a)$$

$$y(i+1) = V \cos \gamma(i) \sin \chi(i) \Delta t + y(i) \quad (5.12b)$$

$$z(i+1) = V \sin \gamma(i) \Delta t + z(i). \quad (5.12c)$$

After reaching the iteration limit of the loyal wingman PSO algorithm identified in the previous chapter, the result is a vector, length  $N$ , of evenly-spaced state elements

$\{x, y, z, \gamma, \chi\}$ . The NLP still requires an initial guess in the form of a vector, length  $N$ , for two additional states,  $\{N_z, \mu\}$  and controls,  $\{u_1, u_2\}$ . This can be done by first algebraically rearranging Equations 5.5d, 5.5e, and 5.5f to

$$\mu(t) = \tan^{-1} \left[ \frac{\dot{\chi}(t) \cos \gamma(t)}{\dot{\gamma}(t) + \frac{g}{V} \cos \gamma(t)} \right] \quad (5.13a)$$

$$N_z(t) = \left[ \frac{1}{\cos \mu(t)} \right] \left[ \frac{\dot{\gamma}(t)V}{g} + \cos \gamma(t) \right], \quad (5.13b)$$

then discretizing Equations 5.13a, and 5.13b, and solving with a backward difference formulation,

$$\mu(i-1) = \tan^{-1} \left[ \frac{\frac{\chi(i)-\chi(i-1)}{\Delta t} \cos \gamma(i-1)}{\frac{\gamma(i)-\gamma(i-1)}{\Delta t} + \frac{g}{V} \cos \gamma(i-1)} \right] \quad (5.14a)$$

$$N_z(i-1) = \left[ \frac{1}{\cos \mu(i-1)} \right] \left[ \frac{\gamma(i) - \gamma(i-1)V}{\Delta t g} + \cos \gamma(i-1) \right], \quad (5.14b)$$

$\forall i = N, N-1, \dots, 2$ . This provides a vector length  $N-1$  for vertical acceleration,  $N_z$  and roll,  $\mu$ . Equations 5.5f, and 5.5g are also discretized and solved using a backwards differencing formulation,

$$u_1(i-1) = \frac{N_z(i) - N_z(i-1)}{\Delta t} \quad (5.15a)$$

$$u_2(i-1) = \frac{\mu(i) - \mu(i-1)}{\Delta t}, \quad (5.15b)$$

$\forall i = N-1, N-2, \dots, 2$  to produce a vector, length  $N-2$  of control inputs,  $u_1$  and  $u_2$ . A discrete, evenly spaced vector for each state and control is now available to supply as an initial guess to the NLP.

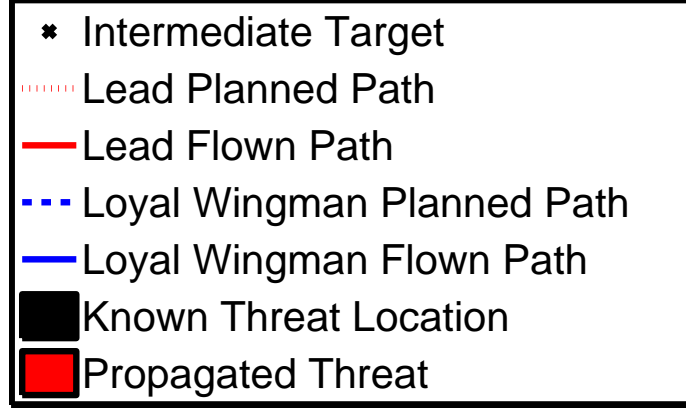


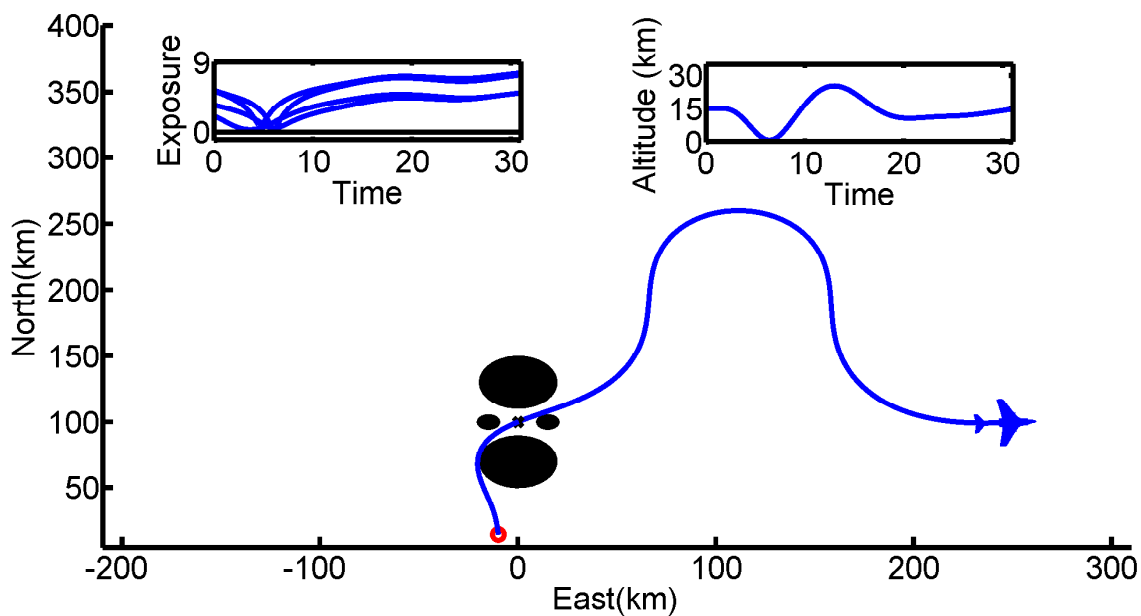
Figure 5.1. Plot Legend for Figures 5.2, 5.3, 5.4, and 5.5

## 5.4 Results

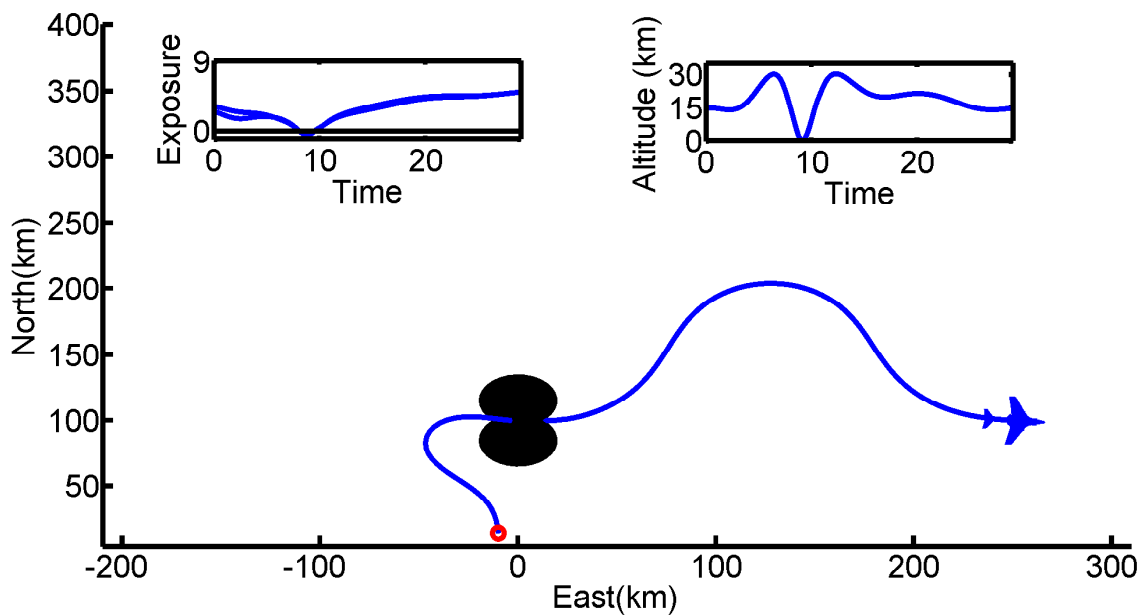
The six identified scenarios, repeated here for convenience, were run to present results for each of the three boundary conditions through both an avoidable and unavoidable threat layout.

1. Fixed time to fixed location with avoidable threats,  $J_{FixedTime}$
2. Fixed time to fixed location with unavoidable threats,  $J_{FixedTime}$
3. Minimize time to fixed location with avoidable threats,  $J_{MinTime}$
4. Minimize time to fixed location with unavoidable threats,  $J_{MinTime}$
5. Minimize time to lead rendezvous with avoidable threats,  $J_{MinTime}$
6. Minimize time to lead rendezvous with unavoidable threats,  $J_{MinTime}$

A legend is provided in Figure 5.1 and is used to describe the plots found in the upcoming results. The starting and intermediate conditions are the red ‘circle’ and black ‘x’, respectively. The loyal wingman’s path is indicated in blue and the lead’s path is marked in red. A dashed line indicates a planned path and a solid line indicates a flown path. Known static threats are indicated as dark shaded regions.



(a) Avoidable Threats



(b) Unavoidable Threats

Figure 5.2. Fixed Time to Fixed Point

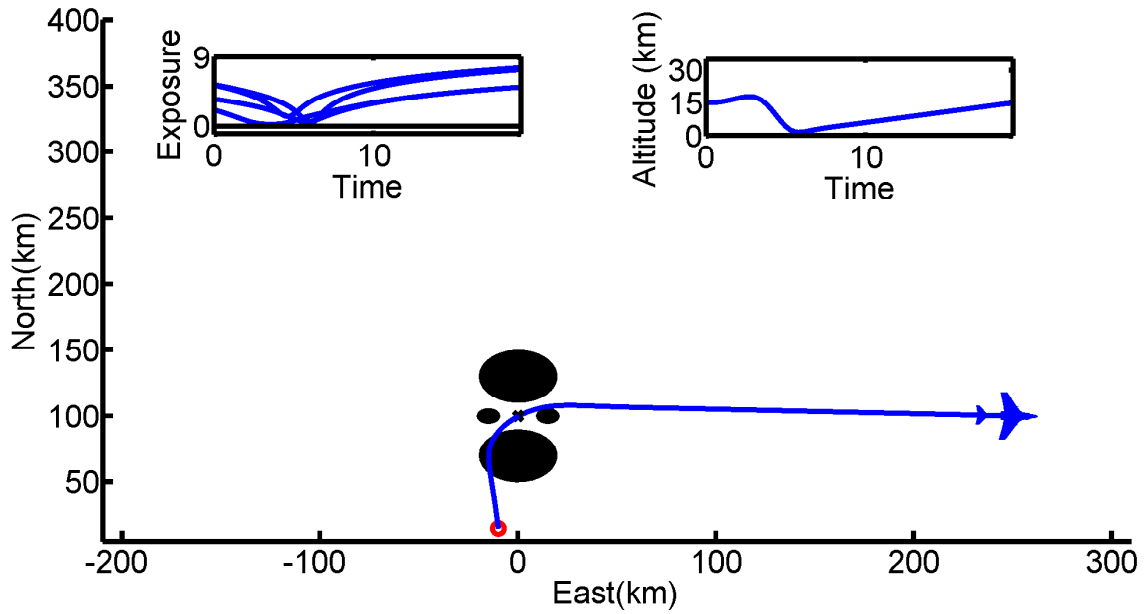
Referring to Figure 5.2, additional quantities are included in two sub-axes. The first axis indicates the loyal wingman's flown path exposure, calculated using the logarithm of Equation 5.2<sup>1</sup>. Values less than zero indicate increased risk of exposure from being inside the threat region. The second axis is a plot of the altitude of the loyal wingman. All altitude plots indicate a starting position of 15km, intermediate target of 2km and final altitude of 15km. In many cases, it may appear from a 2-D display that loyal wingman trajectory is going through a threat, when in reality, the trajectory passes at an altitude higher than the threat's altitude. Figure 2.1 provides an image of the 3-D contour of the superquadric threat keep-out region.

Figure 5.2 includes two results from a fixed-time, fixed-location boundary condition scenario. Subplot (a) is the trajectory through avoidable threats. The loyal wingman trajectory as well as the exposure plot indicate that all threats were successfully avoided. The trajectory also indicates a minimize control idle flight time in order to meet the fixed final time requirement. The computation time for the hybrid method was 25.33 seconds, where the PSO took 18.73 of those seconds. Subplot (b) is the same boundary condition scenario, but there is an unavoidable threat layout due to a fortified target. In order to minimize exposure, the optimizer produces a solution which is the perpendicular bisector of the overlapping threats. A study of the altitude plot indicates that the optimizer additionally took advantage of the altitude to minimize exposure by climbing high and diving into the threat exposure region, where the intermediate target is located. The loyal wingman trajectory then climbs out of the exposure region and flies a serpentine route to meet the fixed final time requirement. The computation time for the hybrid method was 21.83 seconds, where the PSO took 16.58 of those seconds.

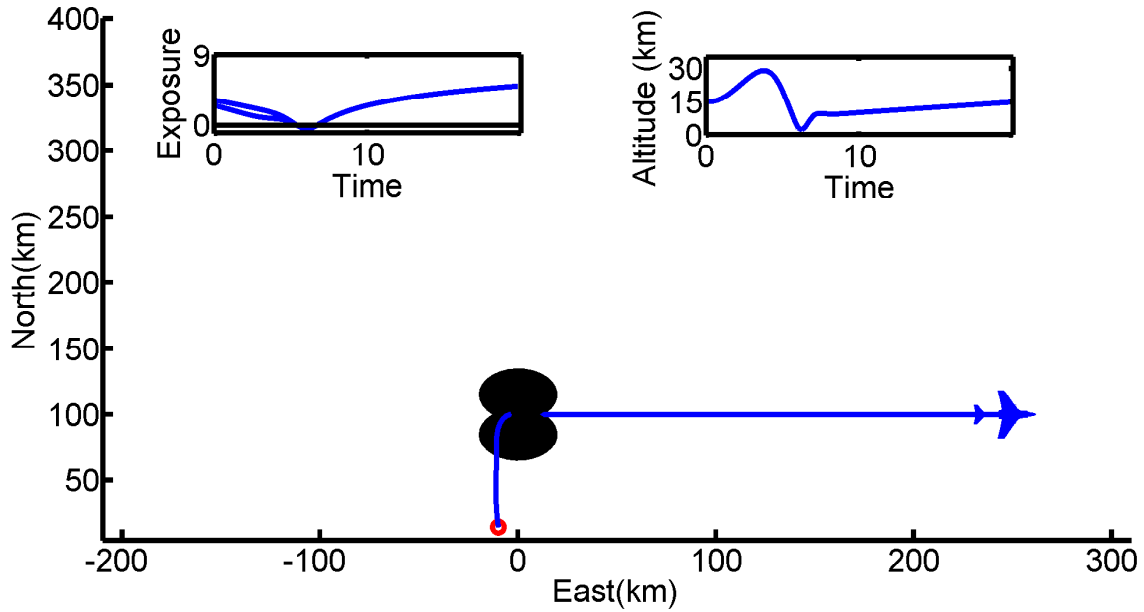
---

<sup>1</sup>Equation 5.2 may grow without bound, so the logarithm is used to scale results for readability.

Figure 5.3 includes two results from the minimize time to a fixed location scenario. Subplot (a) provides the results through an avoidable threat layout. A first glance



(a) Avoidable Threats



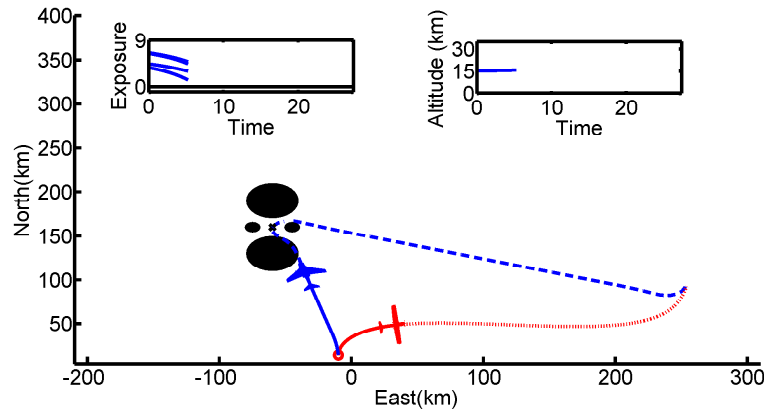
(b) Unavoidable Threats

**Figure 5.3. Minimize Time to Fixed Point**

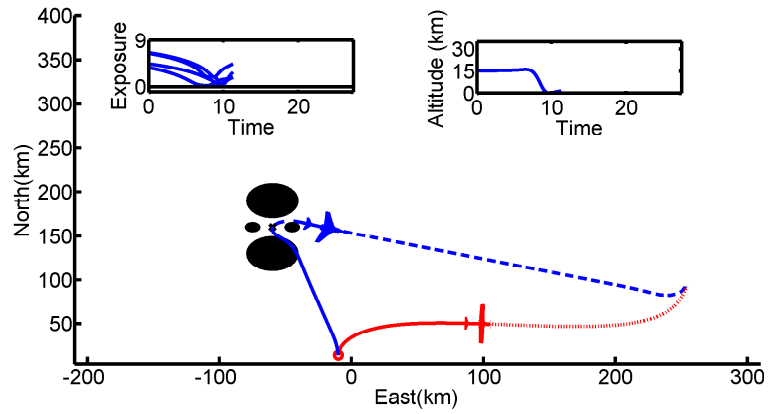
may lead one to believe the trajectory will be exposed to the threat, but a closer look at the exposure and altitude sub-axes shows that the trajectory successfully avoids all

threats by flying at an altitude higher than the threat's altitude. A second property of the resulting trajectory is a direct route to the intermediate and final points in order to minimize time while avoiding threats. The computation time for the hybrid method was 42.68 seconds, where the PSO took 33.68 of those seconds. Subplot (b) is the same minimize time to fixed point boundary condition as above, however this scenario includes an unavoidable threat layout, due to a fortified target. Just as the previous scenario, at first glance it appears the trajectory unnecessarily flies through an exposure region. However, a look at the two sub-axes indicate that the trajectory flies to a high altitude, then dives through the center of the overlapping threats to meet the intermediate target condition. The trajectory then quickly climbs out of the threat exposure region and flies direct to the final position in order to minimize time. The computation time for the hybrid method was 37.66 seconds, where the PSO took 32.87 of those seconds. Figure 5.4 provides the results of minimize time to rendezvous with the manned lead through an avoidable threat layout. In order to highlight the time-dependent final condition associated with lead rendezvous, the scenario is presented in three static subplots (a-c). The trajectory successfully avoids all threats by flying at an altitude above one of the threats. The loyal wingman dives to meet the intermediate target requirement, then climbs directly toward the manned lead's path. The final subplot indicates a trajectory approaching final rendezvous with the manned lead. The computation time for the hybrid method was 34.83 seconds, where the PSO took 22.99 of those seconds.

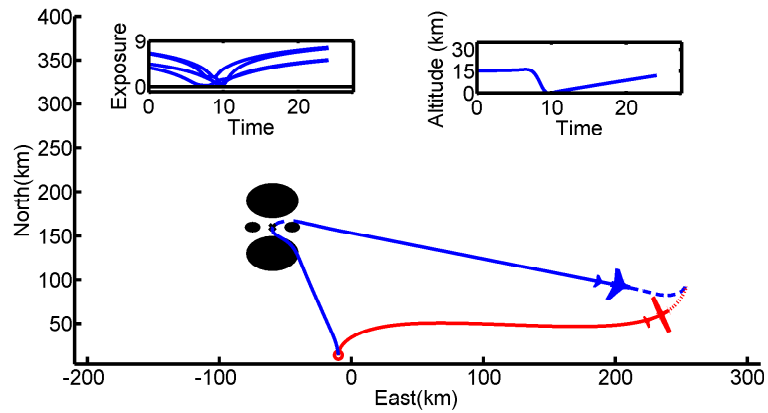
The results of the final scenario, Figure 5.5 are presented in a fashion similar to the previous scenario, as three static subplots (a-c). The results are similar to the minimize time to a fixed location scenario, Figure 5.3, in which the trajectory flies at an altitude higher than the threat, then dives through a perpendicular bisector towards the intermediate waypoint in order to minimize exposure. There is then a



(a) Time = 5.25 minutes

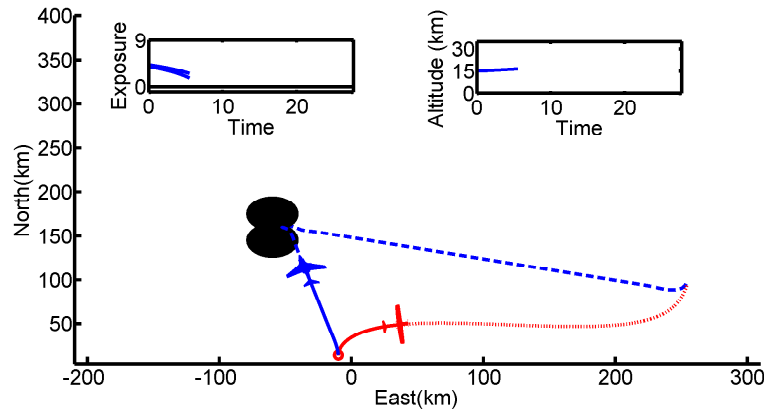


(b) Time = 11.25 minutes

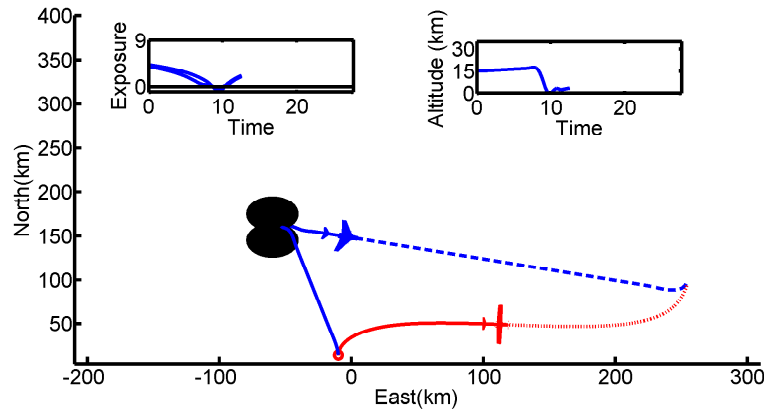


(c) Time = 24 minutes

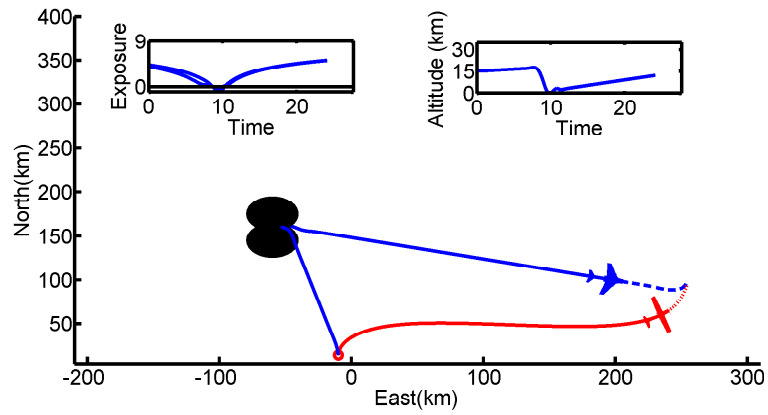
Figure 5.4. Minimum Time to Rendezvous with Lead Through Avoidable Threats



(a) Time = 5.5 minutes



(b) Time = 12.5 minutes



(c) Time = 24 minutes

Figure 5.5. Minimum Time to Rendezvous with Lead Through Unavoidable Threats

swift exit from the exposure region, followed by a direct route to rendezvous with the manned lead. The computation time for the hybrid method was 28.54 seconds, where the PSO took 18.66 of those seconds. Table 5.4 provides a quick look and comparison of computation times for each scenario.

**Table 5.1. Computation Time for Each Scenario**

Scenario	PSO	Hybrid
Fixed Time to Fixed Point Avoidable Threats	18.73	25.33
Fixed Time to Fixed Point Unavoidable Threats	16.58	21.83
Minimize Time to Fixed Point Avoidable Threats	33.68	42.68
Minimize Time to Fixed Point Unavoidable Threats	32.87	37.66
Minimize Time to Lead Rendezvous Avoidable Threats	22.99	34.83
Minimize Time to Lead Rendezvous Unavoidable Threats	18.66	24.54

## 5.5 3-D Model Conclusions and Recommendations

In this chapter, the loyal wingman problem formulation was updated for a 3-D model, adding real-world fidelity to research question one. The loyal wingman PSO was modified in order to produce seeds in three dimensions as well as to compute results using a reduced-order dynamics model. For all scenarios, the NLP optimizer returned feasible solutions meeting the optimality criteria in computation times ranging between 20 and 45 seconds. Overall, the computation time increased from the 2-D model to the 3-D model due to the increase in state variables for both the PSO and the NLP. The computation time of the PSO was offset by the use of the reduced-order model.

As discussed in Chapter IV, additional research should consider continued improvement on the PSO algorithm in order to increase the stochastic search region while at the same time decreasing the overall computation time.

## VI. Dynamic, Non-Deterministic Threats

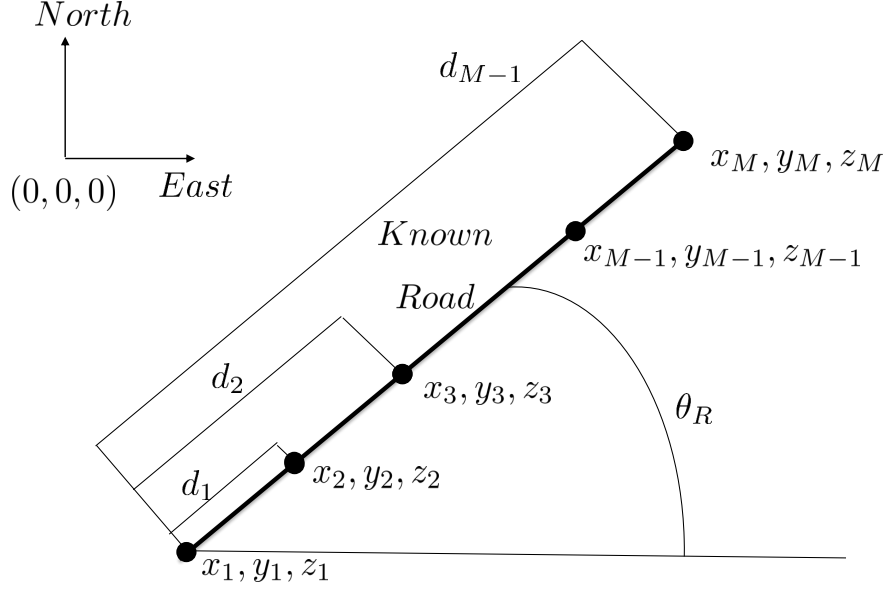
The purpose of this chapter is to answer research question two by developing the model and solving the optimal control problem for moving, stochastic threats. This is done by first establishing assumptions and limitations which drive the development of a dynamics model and a measurement update model. A Kalman filter is used to estimate the threat location and its standard deviation; information which is used to model the threat-keep out region. Finally, a simulation is run which shows a trajectory that successfully avoids the moving, stochastic threat throughout the mission.

### 6.1 Assumptions and Limitations

There are numerous aspects of the problem which may be considered moving and stochastic. It is assumed the loyal wingman itself will have an accurate navigation system and inner-loop controller to maintain the computed trajectory. It is additionally possible that threats as well as the intermediate targets and rendezvous points are moving and stochastic. The work herein will focus on the moving, stochastic threat and suggests the same approach may be applied to intermediate targets and rendezvous points in future research. For simplicity it is assumed the discrete coordinates of a straight road are known *a priori* and that the dynamic threat remains on that road. Variations of this assumption are discussed as topics for future research.

### 6.2 Dynamic Threat Model

Referring to Figure 6.1, the coordinates along a straight road  $(x_R(j), y_R(j), z_R(j))$ ,  $\forall j = 1, 2, \dots, M$  are known *a priori* and the dynamic threat, driving along the road, remains on the road. This allows for the position,  $(x_T, y_T, z_T)$ , of the moving, stochastic threat to be estimated using a 1-D distance-only model. Prior to any estimation,



**Figure 6.1. Coordinates and Distance Along a Known Road**

the azimuth of the road may be computed as

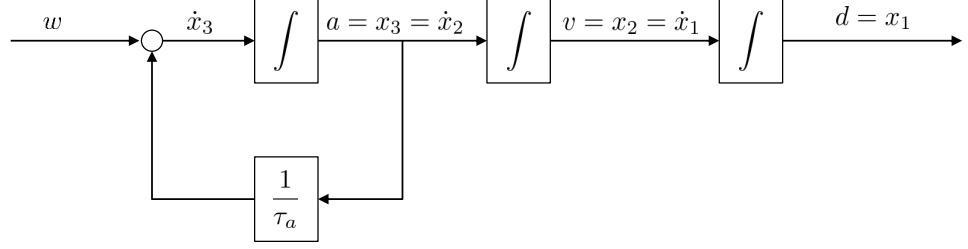
$$\theta_R = \theta_T = \tan^{-1} \left[ \frac{y(M) - y(1)}{x(M) - x(1)} \right] \quad (6.1)$$

and the distance along the road,  $d_R(j)$ , is computed as

$$d_R(j) = \sum_{k=1}^j \sqrt{(dx(k))^2 + (dy(k))^2 + (dz(k))^2}, \quad (6.2)$$

$\forall j = 1, 2, \dots, M-1$ , where  $dx(k) = x_R(k+1) - x_R(k)$ ,  $dy(k) = y_R(k+1) - y_R(k)$ , and  $dz(k) = z_R(k+1) - z_R(k)$ ,  $\forall k = 1, 2, \dots, M-1$ .

Referring to Figure 6.2, the dynamic, non-deterministic distance of the threat along the road is modeled using a stationary first-order Gauss-Markov acceleration with zero-mean, white Gaussian noise strength  $E[w(t)w(t + \tau_a)] = Q\delta(\tau_a)$ , where  $\delta$



**Figure 6.2. Block Diagram of Threat Dynamics Model**

is the Kronecker delta function and  $\tau_a$  is the autocorrelation coefficient. In order to achieve the process noise strength  $\sigma^2$ ,  $Q$  is set to the desired value using  $\frac{2\sigma^2}{\tau_a}$  [144]. Modified from Stankovic [11] and Smith [8] to meet the assumptions of the loyal wingman problem, the discrete linear state-space model is shown in Equation 6.3,

$$\vec{x}_{i+1} = \phi \vec{x}(i) + w(i) = \begin{bmatrix} 1 & \Delta t & \frac{(\alpha \Delta t - 1 + e^{-\alpha \Delta t})}{\alpha^2} \\ 0 & 1 & \frac{(1 - e^{-\alpha \Delta t})}{\alpha} \\ 0 & 0 & e^{-\alpha \Delta t} \end{bmatrix} \vec{x}_i + w_i, \quad (6.3)$$

where  $\alpha = \frac{1}{\tau_a}$  and  $Q$  is discretized to  $Q_d$  through the vanLoan method [130] and the three states represent distance, velocity and acceleration of the threat. Because this is a linear model and Gaussian probability assumptions can be made on the estimate, a Kalman filter is used, as in Brown [130],

$$\hat{x}(i+1) = \phi \hat{x}(i) \quad (6.4a)$$

$$P(i+1) = \phi P(i) \phi^T + Q_d, \quad (6.4b)$$

$\forall i = 1, 2, \dots, N$  time-steps to estimate the distance the threat has traveled  $d_T(i)$  and

a standard deviation,  $\sigma_T(i)$ . A limitation of the practical application of this model is that velocity may grow beyond possible speeds of the threat. Future work should test the model to see if an unbound velocity growth is an issue. If it is an issue, then a recommended variation of the model is to place a limit on velocity growth and, due to the nonlinearity imposed by the limit, utilize an EKF as an estimating tool. All information necessary for modeling the threat and evaluating threat exposure at any given time is now available and when interpolated ensures a continuous and differentiable function: Given a discrete time  $t(i)$  in the loyal wingman's trajectory, the estimated distance the threat has traveled  $d_T(i)$  and standard deviation  $\sigma_T(i)$  may be evaluated. Once the estimated distance traveled is evaluated, it may then be used to compute the distance-parameterized position  $(x_T(i), y_T(i), z_T(i))$ , of the threat along the road.

### 6.3 Modeling Dynamic Threat Avoidance Region

Using superquadrics, the parameters necessary to model the threat region are the global coordinates of the threat center location,  $(x_T, y_T, z_T)$ , the principal axes of the threat,  $(a_x, a_y, a_z)$ , where  $a_x$  is a local coordinate aligned with the road,  $a_y$  is perpendicular to the road and  $a_z$  is vertical, the shape-forming parameter  $p$  and a rotation matrix  $M_{3 \times 3}(\theta_T)$  used to rotate between the local and global coordinate frames.

In the global coordinate frame, the estimated threat position  $(x_T(i), y_T(i), z_T(i))$  at any time  $t(i)$  are determined consistent with the discussion in the previous subsection.

There are two parameters necessary for modeling the size and shape of the threat keep-out region. The size of a deterministic, static threat is established using the principal axes lengths,  $a_x, a_y, a_z$ . The size of the stochastic, moving threat is adjusted by extending length along the axis which the threat is traveling by the standard

deviation,  $\sigma_T(i)$ , such that the x-axis length is

$$a_x(i) = a_x(0) + \sigma_T(i), \quad (6.5)$$

$\forall i = 1, 2, \dots, N$ . The shape of a deterministic static threat is formed based on an initial shaping parameter,  $p(0) = 2$ , to form an ellipsoid. The shaping parameter varies for stochastic, moving threats by

$$p(i) = p(0) + \frac{\sigma_T(i)}{a_x(0)}, \quad (6.6)$$

$\forall i = 1, 2, \dots, N$  to form a cylindrical-like shape whose long axis is extended along the road by Equation 6.5. Additional details on this choice to model the threat keep-out region of moving, stochastic threats is discussed in Appendix D.

### 6.3.1 Computing Threat Exposure.

The purpose of establishing a dynamic threat model is to evaluate discrete points along the loyal wingman trajectory to evaluate threat exposure. A homogeneous transformation matrix may be calculated at discrete time-steps  $t(i)$ ,  $\forall i = 1, 2, \dots, N$ ,

$$\mathbf{R}_{4 \times 4}(i) = \begin{bmatrix} \mathbf{M}_{3 \times 3} & \mathbf{x}_{T_{3 \times 1}}(i) \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix}, \quad (6.7)$$

where

$$\mathbf{x}_T(i) = \begin{bmatrix} x_T(i) \\ y_T(i) \\ z_T(i) \end{bmatrix}, \quad (6.8)$$

to transform the loyal wingman's global coordinates to a threat-centric local coordinate frame by

$$\begin{bmatrix} x_{eval}(i) \\ y_{eval}(i) \\ z_{eval}(i) \\ \sim \end{bmatrix} = \mathbf{R}_{4 \times 4}(i) \begin{bmatrix} x_W(i) \\ y_W(i) \\ z_W(i) \\ 1 \end{bmatrix}, \quad (6.9)$$

through which the exposure at each discrete point in the loyal wingman's trajectory may be evaluated using Equation 6.10,

$$F(i) = \left( \frac{x_{eval}(i)}{a_x(i)} \right)^{p(i)} + \left( \frac{y_{eval}(i)}{a_y} \right)^2 + \left( \frac{z_{eval}(i)}{a_z} \right)^2. \quad (6.10)$$

The special form of Equation 6.10 represents a 'growing' threat region along the x-axis.

### 6.3.2 Measurement Update Model.

It is expected that when the optimal path is computed, the initial condition of the non-deterministic, dynamic threats will be known and the propagate portion of the Kalman filter will be run. The loyal wingman will compute the optimal path to complete the mission while avoiding or minimizing exposure to the threats throughout the mission, recognizing that the threat keep-out region changes at each time step.

After the original path is planned, the loyal wingman begins its flight along the computed path, however an assumed on-board sensor provides updated Cartesian coordinates of the threat location with a known margin of error at each discrete time-step. The distance along the road associated with the sensor's measurement is determined by table lookup from the road coordinates known a priori. The measurement model is then

$$z(i) = h(x(i)) + \nu(i) \quad (6.11)$$

where  $\nu(i)$  is zero-mean white Gaussian noise with strength  $E[\nu(t)\nu(t)] = R\delta$ , where  $R$  indicates the uncertainty associated with the sensor. With the values  $H = h = 1$  and  $R = r = \sigma_e^2$ , the Kalman filter measurement update equations are calculated in combination with Equation 6.4 found in Brown [130],

$$K(i) = P^-(i)H^T [HP^-(i)H^T + R]^{-1} \quad (6.12a)$$

$$\hat{x}(i) = \hat{x}^-(i) + K(i)[z(i) - H\hat{x}^-(i)] \quad (6.12b)$$

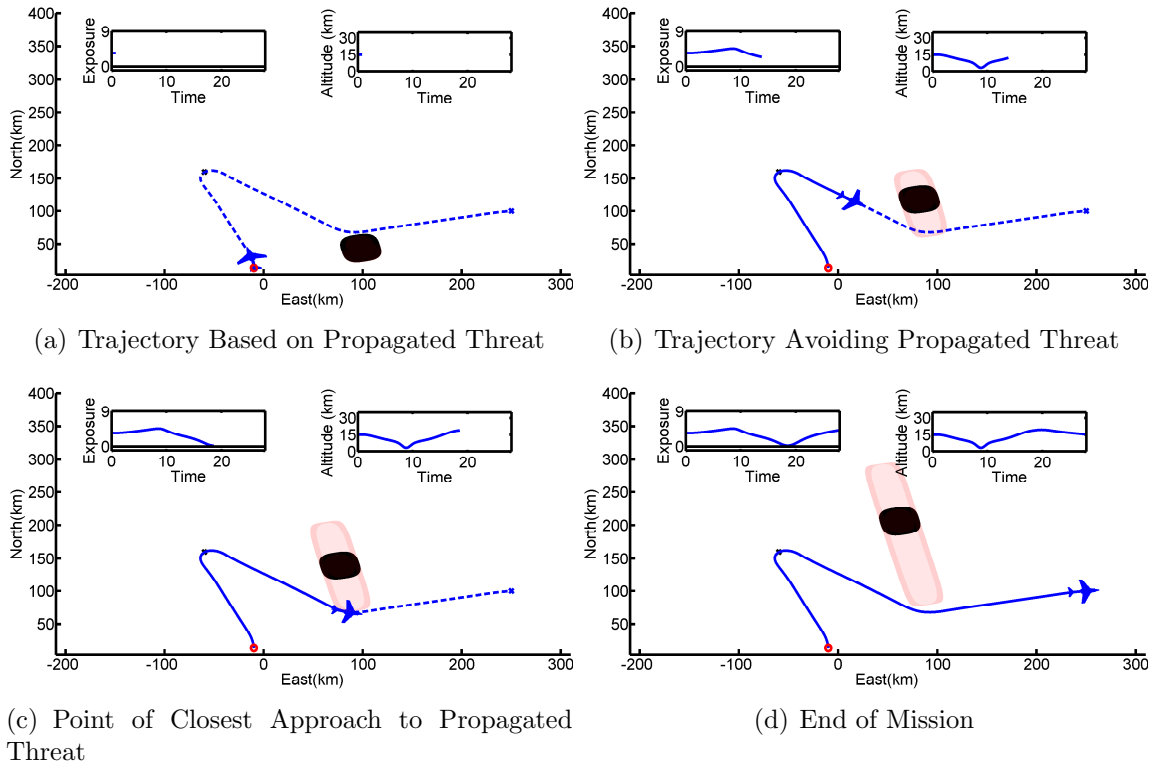
$$P(i) = [I - K(i)H]P^-(i). \quad (6.12c)$$

Measurement updates may be provided at an identified desired rate which, when used with the Kalman filter, will provide a better estimate of the distance a threat may have traveled. As measurement update rates increase, the estimation accuracy improves.

## 6.4 Results

Two scenarios are simulated whose results demonstrate the tools discussed in this chapter. Figures 6.3 and 6.4 show the results of two scenarios, 1.) minimize time to fixed location, and 2.) minimize time to rendezvous with manned lead. Both scenarios include a moving, dynamic threat that must be avoided. Plot properties are the same as discussed in Chapter V. The starting, intermediate and final condition points are the red ‘circle’, black ‘x’, and blue ‘star’, respectively. The loyal wingman’s path is indicated in blue and the lead path in the appropriate scenario is marked in red. A dashed line indicates a planned path and a solid line indicates a flown path. Known static threats are indicated as dark shaded regions. In subsequent subplots, the red shaded region indicates the propagated threat keep-out region, which changes in both size and shape, according to Equations 6.5 and 6.6 at each time-step. Referring

to Figure 5.2, additional plot properties include two sub-axes. The first inset plot indicates the loyal wingman's flown path exposure, calculated using the logarithm of Equation 5.2<sup>1</sup>. Values less than zero indicate increased risk of exposure. The second inset of subplots indicates the altitude of the loyal wingman. All altitude plots indicate a starting position of 15km, intermediate waypoint of 2km and final altitude of 15km, matching the specified constraints. In many cases, it may appear from a 2-D display that the loyal wingman trajectory is going through a threat, when in reality, the trajectory passes at an altitude higher than the threat altitude (as shown in the exposure inset subplot remaining greater than zero). Figure 2.1 provides an image of the 3-D contour of the superquadric threat keep-out region. The DOC method



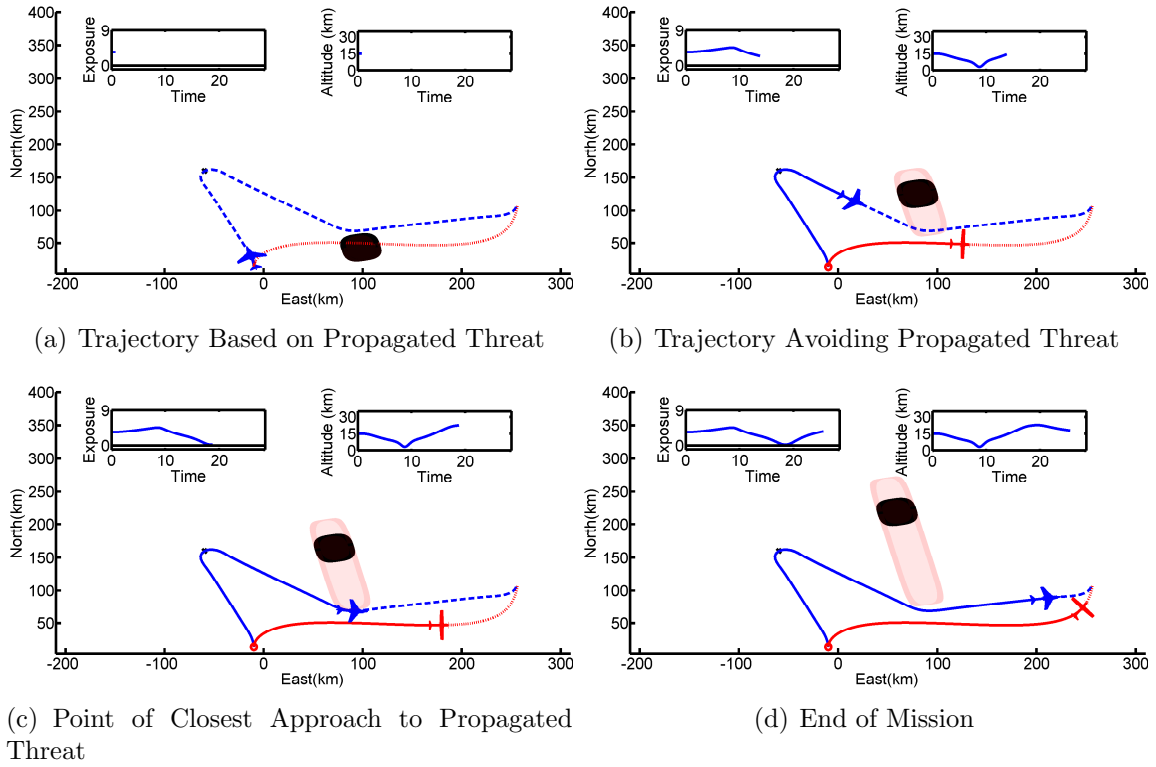
**Figure 6.3. Trajectory Simulation to Avoid Moving, Stochastic Threat with Fixed Point Rendezvous**

used 40 nodes in each of two phases, for a total of 80 nodes. Figure 6.3 represents

<sup>1</sup>Equation 5.2 may grow without bound, so the logarithm is used to scale results for readability.

the minimize time to fixed location scenario. Subplot (a) indicates the trajectory prior to beginning the simulation. The computed trajectory is based on avoiding the threat throughout the mission. Subplot (b) provides a view of the threat situation after the loyal wingman has flown approximately half of the mission. As can be seen, the propagated threat keep-out region is much larger than the threat keep-out region indicated by the measurement update *at that time-step*. This subplot also reveals that a dynamic re-plan of the trajectory may improve the current planned route, which is a motivation for the dynamic re-plan research discussion in the next chapter. Subplot (c) displays the point of closest approach to the propagated threat region. In the final subplot (d), the loyal wingman has surpassed the threat and rendezvoused at the specified fixed final position in minimum time. The computation time for this scenario included 25.6 seconds for the PSO and 6.9 seconds for DOC method, for a total of 32.5 seconds. Figure 6.4 represents minimize time to rendezvous with the lead. Here, the trajectory of the manned lead is known, and the loyal wingman must overfly the target, avoid the threat and rendezvous (match position and heading) in minimum time. The four subplots are similar to the discussion above only the final boundary condition is changed to reflect rendezvous with the manned lead in minimum time. The computation time for this scenario included 21.9 seconds for the PSO and 6.1 seconds for DOC method, for a total of 28 seconds.

As a final point of discussion for this section, this work uses the GPOPS II Matlab-based software to transcribe the optimal control problem into an NLP, which includes establishing a first and second derivative matrix. Patterson and Rao [145] discuss the creation of these matrices and describe a method to exploit their sparseness for improved computation time. The method is provided as an option in the GPOPS II software, which worked well for a static threat environment. However, there were challenges with convergence when using this option in a dynamic threat environment.



**Figure 6.4. Trajectory Simulation to Avoid Moving, Stochastic Threat with Lead Rendezvous**

A review of Patterson and Rao [146] may provide insight on the cause of this challenge. The sparsity exploitation tool was therefore not utilized in the loyal wingman optimal control problem, resulting in no noticeable difference in computation time in a static threat environment, while achieving feasible solutions in the dynamic threat environment. Additional information on the sparsity of matrices can be found in Patterson and Rao [145] and the method used by the GPOPS II software for exploiting the sparsity is described in Patterson and Rao [146].

## 6.5 Non-Deterministic Threat Conclusions and Recommendations

This chapter answered research question two by developing a dynamic and measurement update model as well as the use of a Kalman filter to estimate the location of a moving, stochastic threat. The keep-out region of the stochastic threat is modeled using two parameters. The first is a time-dependent superquadric shaping parameter to shape the threat keep-out region as a cylinder. The second defines the size of the cylindrical shape by extending its long axis along the axis of the road according to the estimated threat speed. A homogeneous transformation matrix is used to transform the loyal wingman's current location in global coordinates to a local coordinate frame for evaluating the trajectory exposure. A simulation was run and results plotted which show the growth of the threat keep-out region as well as the generated path which successfully avoids the expanded threat keep-out region throughout the mission. The next chapter will introduce a time to re-plan formulation that will re-compute the trajectory during mission flight, based on updated threat information.

The objective was to model the moving, stochastic threat keep-out region and provide results showing the hybrid methodology can provide a rapid, feasible solution. In order to do so a number of simplifying assumptions were made. Future research should consider variations to the assumptions of this chapter. First, the road may

be considered non-straight, suggesting a threat region may be modeled using Smith's SLIMVEE algorithm to create non-straight threat avoidance corridors [8]. Second, the linear 1-D model may be extended to two or three dimensions and as long as the model is linear and Gaussian probability assumptions are made, the Kalman filter may be utilized to estimate location. In addition to increasing the dimension of the threat stochastics model, the current model allows for a velocity estimate that may grow beyond possible speeds of the threat. Future work should test the impact of the current velocity model and test the use of a nonlinear limit-imposed velocity model. Because dynamic threat estimation is computed outside of the NLP, changes in the model have minor impacts on the gradient-based NLP solver. Changes in the dynamic model can be addressed by using other available and proven estimating tools such as the extended Kalman, unscented Kalman and particle filter.

## VII. Dynamic Re-Plan

The purpose of this chapter is to answer research question three by determining if and/or when a dynamic re-plan of the loyal wingman trajectory is necessary. There are two scenarios that will be considered. The first scenario assumes an original trajectory has been computed, and the mission is being flown, when an intelligence source provides updated information indicating a change in the mission environment such as a ‘pop-up’ threat or a change in rendezvous criteria. A trajectory re-planning algorithm must determine whether this change in the mission environment warrants a re-plan and generally will result in an immediate re-plan of the loyal wingman trajectory to meet the updated mission requirements. The second scenario for trajectory re-planning considers the presence of a moving, stochastic threat. For such a scenario, the work herein develops an algorithm to determine *if* a re-plan is necessary, and subsequently, a time to re-plan,  $t_r$ , based on a critical distance metric,  $\rho$ , that uses the assumed speed and stochastics of the threat, as well as relative distance and angle of approach to the threat.

### 7.1 Mission Flow and Changes in Mission Environment

The overall mission re-planning flowchart is shown in Figure 7.1. The left side represents computation of the loyal wingman trajectory, and in the presence of moving, stochastic threats, recommends a time to re-plan the loyal wingman trajectory,  $t_R$ . The right side represents a mission flight, which steps through the trajectory at discrete time-steps, evaluating whether a mission re-plan is necessary based on ‘pop-up’ threats or mission changes. Each step in the flowchart is numbered to facilitate discussion.

The algorithm is initialized (box 2) with parameters passed from the manned lead

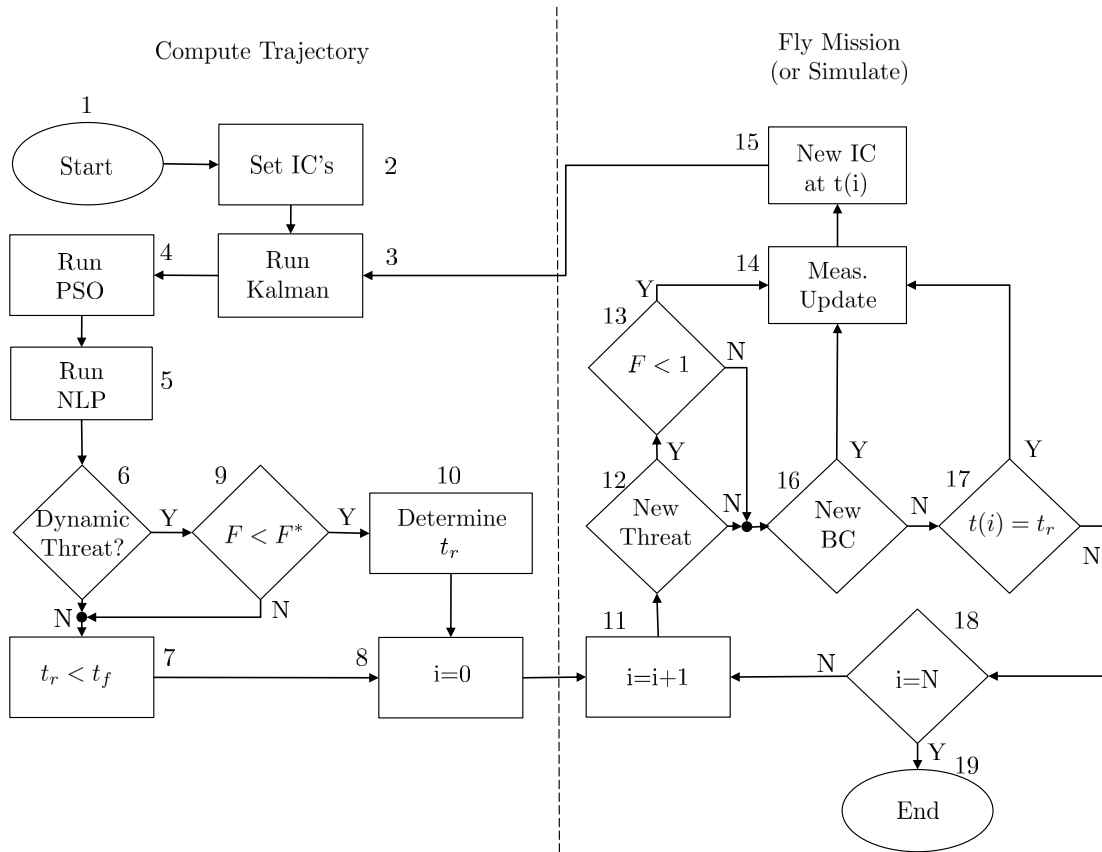


Figure 7.1. Dynamic Re-Planning Flowchart

or other intelligence sources. With parameters initialized, bounds are established and if there is a dynamic threat, the distance traveled by the threat  $d_T$  and a standard deviation  $\sigma_T$  are estimated through a Kalman filter propagation (box 3), as discussed in Chapter VI. This information is passed into a Particle Swarm Optimization (PSO) (box 4) algorithm [147] and the optimal control problem is transcribed into a nonlinear program (NLP). The output from the PSO is provided as an initial guess into the NLP solver and an initial outer-loop trajectory is computed (box 5). If a dynamic threat is present (box 6), then an algorithm (boxes 9 and 10), described in the next section, determines if and subsequently when ( $t_R$ , time to re-plan) the loyal wingman's trajectory should be re-planned. The mission flow then moves from trajectory computation on the left side of Figure 7.1 to mission flight on the right side.

At each discrete time-step, there is a check to see if there is a change in the mission environment. If a new threat has 'popped-up' (box 12), then the loyal wingman's discretized trajectory from the current time forward is evaluated, via Equations 6.5, 6.6, and 6.10, to determine if the new threat will cause exposure. If at any discrete point in the trajectory, the value of the inside-outside function,  $F$ , is less than one, indicating a point in the interior of the threat region, then the loyal wingman risks exposure and the current path should be re-planned (box 13). If any  $F$  is greater than one, then no trajectory update is needed. In addition to pop-up threats, there is a check to see if there is a change in the mission requirements (box 16), such as a new intermediate target or new final rendezvous, as well as a check to see if the current time  $t(i)$  has reached the previously determined time to re-plan  $t_r$  (box 17).

If any of the checks (boxes 13, 16, 17) indicate that a re-plan of the loyal wingman trajectory is necessary, a sensor measurement update on the current location of the threat is taken (box 14), new initial conditions are established (box 15), and the process flows back to the left side of Figure 7.1 (box 3). The entire process of computing

and flying the trajectory is repeated until the mission (or simulation) is complete.

## 7.2 Moving, Non-Deterministic Threats: Time to Re-Plan, $t_r$

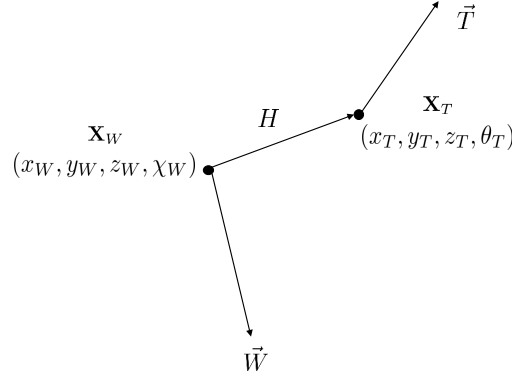
This section describes Figure 7.1, boxes 9 and 10 to determine if and/or when a time to re-plan,  $t_r$ , should occur in the presence of a moving, non-deterministic threat.

### **Determine *IF* a Re-Plan is Necessary.**

Just because there is a threat present (stationary or moving), does not mean the computed trajectory is impacted by the presence of the threat. As an example, if the loyal wingman's mission is to the east and the threat is in the west (and in the case of a dynamic threat, moving west), then the presence of a threat would not impact the computed trajectory. Therefore, the algorithm must first determine whether the originally computed trajectory is impacted by the presence of the threat. This is done by evaluating Equations 6.5, 6.6, and 6.10 at all time-steps of the loyal wingman's trajectory. If the threat exposure at any discrete point is below an established threshold  $F(i) < F^*$ , evaluated  $\forall i = 1, 2, \dots, N$  then the loyal wingman's trajectory is in the vicinity of the threat. As described in Chapter III, depending on the user choice of the stiffness value  $s$ , it is likely the optimizer will converge on a trajectory in which  $F > 1$ .

### **Determining a Time to Re-Plan, $t_R$ .**

After determining that the computed loyal wingman trajectory was influenced by the presence of a threat, a time to re-plan must be computed. If a loyal wingman trajectory re-plan occurs too early, then little has changed in the threat scenario and the newly computed trajectory will have changed an insignificant amount from the



**Figure 7.2. Angle of Approach ( $\alpha$ ) to Threat Vector Normal**

previous trajectory. However, if the suggested time to re-plan is much later it could place the loyal wingman very close to or past the threat, in which case there is no longer a need to update the trajectory. It is therefore imperative to determine a time to re-plan that considers the speed and stochastics of the moving threat as well as the relative distance and angle of approach of the loyal wingman to the threat.

Referring to Figure 7.2, this work develops the critical distance metric,  $\rho$  (defined in Equation 7.4), suggesting a re-plan should occur when the loyal wingman's relative distance to the threat,  $H(i)$ ,

$$H(i) = \|\mathbf{x}_W(i) - \mathbf{x}_T(i)\|_2, \quad (7.1)$$

is less than  $\rho$ . The time to re-plan,  $t_r$ , can then be determined by

$$t_r = i_{min} \Delta t \quad (7.2)$$

where

$$i_{min} = \min\{i : H(i) \leq \rho(i)\} \quad (7.3)$$

evaluated  $\forall i = 1, 2, \dots, N$ . The critical distance metric,  $\rho$ , is developed using Equation 6.5, which models the size of the threat keep-out region at each time-step by extending the axis along the road by the standard deviation at that time-step. Using Equation 6.5 alone, the time to re-plan would occur when the loyal wingman's trajectory coincides with the threat keep-out region in a direct, 'head-on' approach. Because the loyal wingman's trajectory was calculated based on minimizing exposure, it is unlikely this distance will be achieved in a direct approach. Therefore, the minimum length of the principal axis of the threat region, Equation 6.5, is extended by a scaled standard deviation,  $\xi(i)\sigma(i)$  leading to the critical distance metric,  $\rho(i)$ , defined as:

$$\rho(i) = a_x(0) + \sigma_T(i)[1 + \xi(i)], \quad (7.4)$$

where  $\xi(i)$  represents a speed, distance, and angle of approach ratio, defined as:

$$\xi(i) = \left[ \frac{\|\mathbf{x}_T(i) - \mathbf{x}_T(0)\|_2}{H(i)} \right] \left| \cos \alpha(i) \right|. \quad (7.5)$$

There are three factors used to develop  $\xi(i)$  in Equation 7.5, which are now discussed.  $\|\mathbf{x}_T(i) - \mathbf{x}_T(0)\|_2$  represents the distance the threat has traveled from the start of the mission or the last re-plan. A fast threat will trend  $\xi$  to a higher value, whereas a slow threat will tend to keep  $\xi$  small, which has a direct relationship to the increase or decrease of  $\rho$  in Equation 7.4.

The second factor in Equation 7.5 is the relative distance from the loyal wingman to the estimated threat position,  $H$ , defined in Equation 7.1. Using an inverse relationship, if the relative distance decreases, then the likelihood of triggering a re-plan increases. If, on the other hand,  $H$  increases, the likelihood of triggering a re-plan

decreases.

The third factor in Equation 7.5 is the relative angle of approach, calculated as the dot product of the velocity vectors between the wingman and the center of the threat region,

$$\cos \alpha(i) = \frac{\vec{W}(i)^T \vec{T}(i)}{||\vec{W}(i)|| ||\vec{T}(i)||}. \quad (7.6)$$

Equation 7.6 produces values on the interval  $[-1, 1]$ . A value of either  $-1$  or  $1$  indicates a ‘head-on’ approach. It does not, however, indicate whether the vehicles are traveling toward each other (decreasing  $H$ ) or away from each other (increasing  $H$ ). Therefore Equation 7.5 takes the absolute value such that the interval is  $[0, 1]$ . With this formulation, a ‘head-on’ approach scales  $\xi$  to 1 and a perpendicular approach scales  $\xi$  to 0.

It is assumed the speed of the loyal wingman is much greater than the speed of the threat. As a result, after the loyal wingman has overcome the threat, the distance,  $H$ , increases at a rate fast enough that Equation 7.2 is not satisfied and no re-plan is triggered.

Appendix E provides a simulation that demonstrates the effects relative distance (Equation 7.1), the critical distance metric (Equation 7.4), relative speed and distance ratio (Equation 7.5), and angle of approach (Equation 7.6) have on determining a time to re-plan.

### 7.3 Results

A scenario was established which included using the hybrid optimization technique to solve the optimal control problem as well as using the techniques discussed throughout this work to dynamically re-plan the optimal control problem when a non-deterministic, moving threat ‘pops-up’. Referring to Figure 7.3, the plotting properties are identical to the description of Chapter VI. Subplot (a) provides an initially

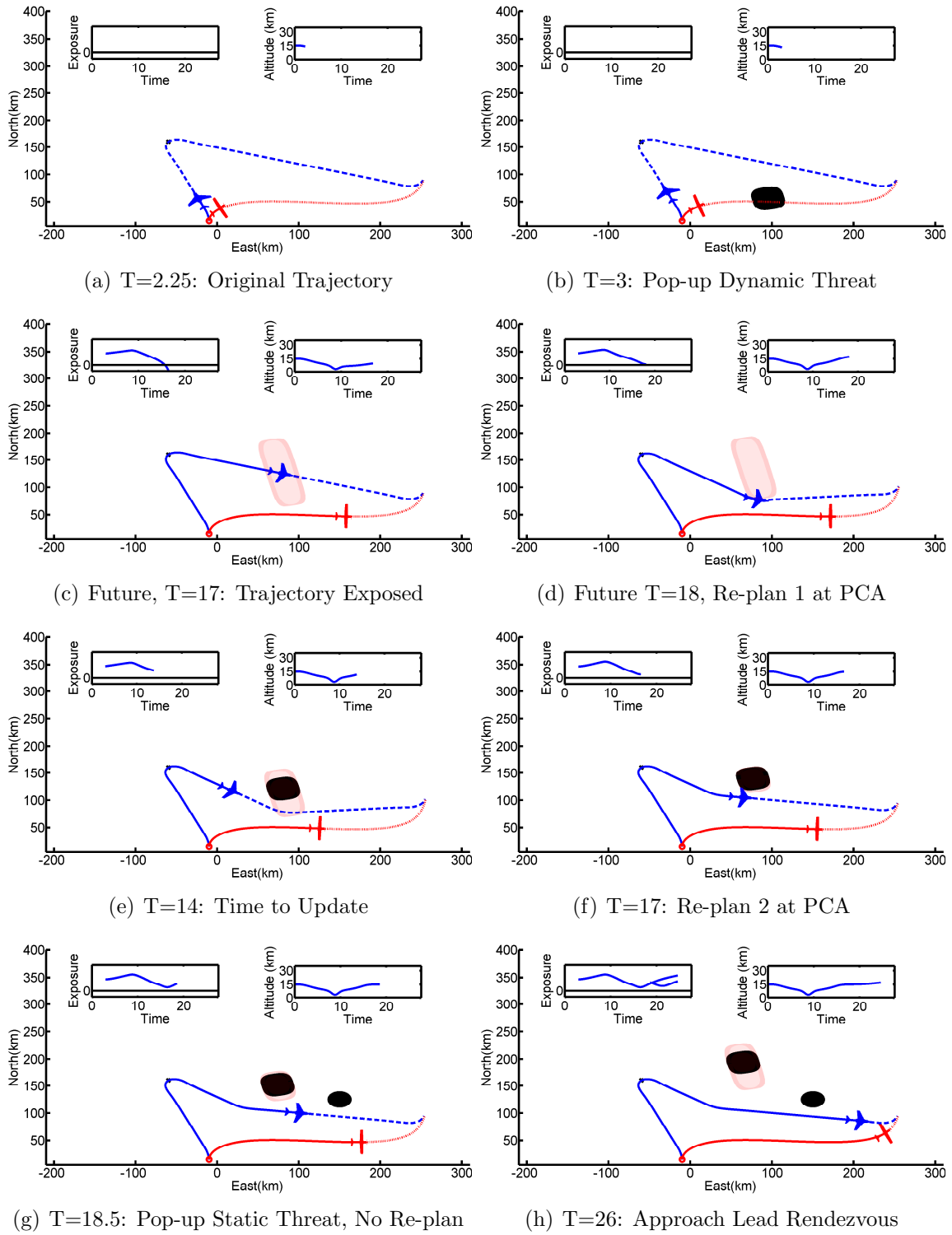


Figure 7.3. Demonstration of Re-Plan in Changing Mission Environment

computed trajectory for the loyal wingman to break from formation, accomplish the intermediate target and rendezvous with the manned lead. In subplot (b), a dynamic threat ‘pops-up’ and although it may appear the trajectory will not be exposed, once the threat keep-out region has been propagated, a future look in subplot (c) reveals, there is a risk of exposure. Therefore, a re-plan must occur. A new trajectory is calculated and subplot (d) provides a future look of the newly computed trajectory, revealing the risk of exposure is low at the point of closest approach (PCA) to the propagated threat. When the re-plan occurs at  $T = 3$ , the algorithm additionally computes a time to re-plan,  $t_r = 14$ . The loyal wingman flies until  $T = t_r$ , subplot (e), and a second re-plan occurs. The new trajectory is shown in subplot (f) at the point of closest approach to the propagated threat. In subplot (g), a static threat ‘pops-up’, but because it does not pose a threat to the loyal wingman’s trajectory, no re-plan occurs. Finally, in subplot (h), the loyal wingman is in its final approach for rendezvous with the manned lead.

## 7.4 Dynamic Re-Plan Conclusions

A mission re-planning flowchart was provided in Figure 7.1 and fully described, which includes a formulation to determine *if* a re-plan is necessary as well as an appropriate time to update the loyal wingman’s trajectory *when* the mission scenario includes moving, non-deterministic threats. The time to re-plan is computed after the loyal wingman’s trajectory has been established, prior to flight, and is determined by the critical distance metric developed herein, which used assumed threat speed and stochastics as well as the loyal wingman’s relative distance and angle of approach to the threat. The work is demonstrated through simulation and shows a successful minimum time rendezvous with the manned lead. The simulation additionally demonstrates the ability to adjust the original trajectory when a dynamic

threat pops-up, as well as update the trajectory according to the critical distance metric. A natural extension of this work may consider re-planning in the midst of multiple dynamic threats as well as additional intermediate targets. Additionally, the time to re-plan,  $t_r$ , is computed with no consideration to computation time. A more accurate representation of the computation time will be known after flight test hardware has been determined and the system flown. Future work should reduce the time to re-plan,  $t_r$ , by the estimated computation time of the algorithm.

## VIII. Conclusions and Recommendations

This work sought to contribute to the scientific and engineering community by developing a methodology and set of research questions that, when answered, would aid in developing the DoD concept of MUM-T, a subset of which is the loyal wingman. A specific definition and candidate scenarios for the loyal wingman were established to appropriately scope the research. It was suggested that optimal control and stochastic estimation techniques could be used to solve the loyal wingman problem *as defined herein*. A methodology was proposed and a set of research questions generated, which, when answered, successfully demonstrated the methodology and provided a contribution to the body of knowledge. The research also led to key findings and recommendations highlighted in this chapter.

### 8.1 Research Questions

1. **The optimal control problem was formulated and solved for a static threat environment.** The optimal control problem was formulated using both 2-D and 3-D models. The problem was divided into two phases and solved using direct orthogonal collocation. Multiple boundary condition scenarios were established, including fixed time to fixed location, minimize time to fixed location, and minimize time to rendezvous with time-dependent manned lead's path. Threats were modeled as superquadrics due to the low number of parameters necessary for generating various sizes and shapes which aids in maintaining a low communication bandwidth as well as computational efficiency. In order to account for both avoidable and unavoidable (target within threat region) threat scenarios, the threats were included in the cost functional to minimize time of exposure using a modified inside-outside product function in addition to

time and control. To ameliorate the issue of convergence to local minimums, a hybrid optimal control methodology was demonstrated which used a heuristic-based optimization technique to generate an initial guess. The guess was then supplied to the direct orthogonal collocation (DOC)-transcribed gradient-based nonlinear programming (NLP) solver to provide a rapid, accurate solution.

2. **The optimal control problem was formulated and solved for a moving, stochastic threat environment.** A 1-D linear dynamic and measurement update model was developed to mimic a threat traveling along a straight road whose location and standard deviation were estimated using a Kalman filter and modeled using a superquadric. The size and shape of the threat region is formed based on the time-dependent standard deviation. The size is varied by extending the axis along which the threat is traveling by the standard deviation and the shape is varied from ellipsoidal to cylindrical by varying the superquadric shaping parameter by the standard deviation as well. Using these models, the optimal control problem was solved to ensure the loyal wingman trajectory minimized threat exposure throughout the mission when the location of the threat *at all time-steps* is not known.
3. **A method was developed for determining *if* and *when* the trajectory should be re-planned in a changing mission or dynamic threat environment.** A mission-flow diagram was generated which includes 1.) generation of flight trajectory using methods resulting from research questions one and two, and 2.) mission flight (or simulation). After a trajectory is computed, if a dynamic threat is present and its presence impacts the computed trajectory, a time to re-plan is generated based on the critical distance metric which uses threat speed, direction, and stochastics as well as the loyal wingman's relative distance and angle of approach to the threat. After the trajectory is computed,

the mission is flown. Changes in the mission environment, such as pop-up threats are evaluated for exposure and if necessary the trajectory is re-planned immediately. When the mission flight time coincides with the time to re-plan, a measurement update is taken, initial conditions are re-established and a new trajectory is computed. The cycle of computing a trajectory and then flying the trajectory is repeated until the mission is complete.

Collectively, this work demonstrated a methodology for solving the loyal wingman optimal control problem *as defined herein* in a near real-time environment. This optimal control methodology may also be used as a comparison to evaluate the performance of other methods.

## 8.2 Contributions

In the course of answering the research questions, several specific research contributions have been identified.

1. A method was developed to formulate and solve the optimal control problem for multiple scenarios, including avoidable and unavoidable threats as well as multiple boundary condition scenarios, including fixed time to fixed location, minimize time to fixed location, and minimize time to rendezvous with the manned lead.
2. A method was developed to model threats which is useful for both avoidable and unavoidable threat scenarios. Threats were modeled and formulated in the cost function using the modified inside-outside product function necessary for use in the gradient-based nonlinear programming solver.
3. A method was developed for estimating the location of a moving, stochastic threat. A 1-D model was established and estimation was then used to model

the changing size and shape of the threat keep-out region *at all time-steps*.

4. A method was developed for determining a time to re-plan formulation in the presence of a moving, stochastic threat. The time to re-plan is generated based on the critical distance metric which uses threat velocity and stochastics as well as the loyal wingman's relative distance and angle of approach to the threat.
5. A hybrid optimal control methodology was developed and demonstrated to provide rapid and accurate optimal solutions for the loyal wingman application which may be applied to other applications as well.

### **8.3 Future Research Recommendations**

In addition to identifying contributions, the research uncovered a number of key findings relevant for future research.

#### **8.3.1 Improve the Particle Swarm Optimization Algorithm.**

1. The research herein showed that the hybrid methodology could provide a rapid, accurate solution. Given that the PSO has numerous 'knobs' and many variations suggested in literature, future research should focus solely on developing a PSO algorithm that simultaneously increases the search region and converges toward a global region in a more timely fashion. Determining a good initial guess for a gradient-based solver continues to be an elusive challenge for the optimal control community and the PSO could be an answer to that challenge.
2. The work herein defined the PSO particles as a vector of discrete control inputs. Future work should consider defining the PSO particles differently. Many works in literature define the PSO particles as the coefficients to a polynomial. If this were done in the loyal wingman PSO, a correct polynomial basis set that aligned

itself with the GPOPS II DOC transcription could create synergy between the two methods that may aid in a more rapid solution.

### **8.3.2 Extend the Moving, Stochastic Threat Model.**

1. The work herein established a simplifying assumption that the road was straight. Future research should consider a road that curves. The threat modeling in this case would remain 1-D, but consideration would be given to modeling a threat keep-out regions with a curved primary axis. This can be done by parameterizing the trajectory and extending the shape based on the tangent to the curve or through the use of Smith's SLIMVEE algorithm [8] to establish no-fly corridors.
2. The assumption of a single dimension may be extended to 2-D or 3-D. The threat model would simply extend additional superquadric axes by the standard deviation in the appropriate dimension. As long as the dynamic model remains linear, the Kalman filter may be used. If the dynamic, stochastic model becomes nonlinear or Gaussian assumptions cannot be made, there are other estimation filters which may be utilized.

### **8.3.3 Improve Robustness.**

1. Specific threat and boundary condition scenarios were chosen and demonstrated in order to highlight the results of the methodology for avoiding threat, minimizing exposure, avoiding propagated moving threats, and re-planning to improve optimization objectives. Although the method presented here is prescriptive and was designed to work in multiple threat and boundary condition layouts, it has not been fully tested. Future work should test the robustness of the method to random placement of threat and boundary condition scenarios.
2. The work herein developed a methodology for planning and dynamically re-

planning a trajectory. Many assumptions were made on other technologies which provide inputs to and accept outputs from the loyal wingman route-planning algorithm. The algorithm used for the research herein which provided a method for solving the optimal control problem should be re-built, hosted on an appropriate system interface, and flown in a system flight test. A more accurate representation of the computation time will be known after flight test hardware has been determined and the system flown. A flight test will additionally allow for evaluation of the performance of the method chosen for this work and allow for identification of areas for additional research.

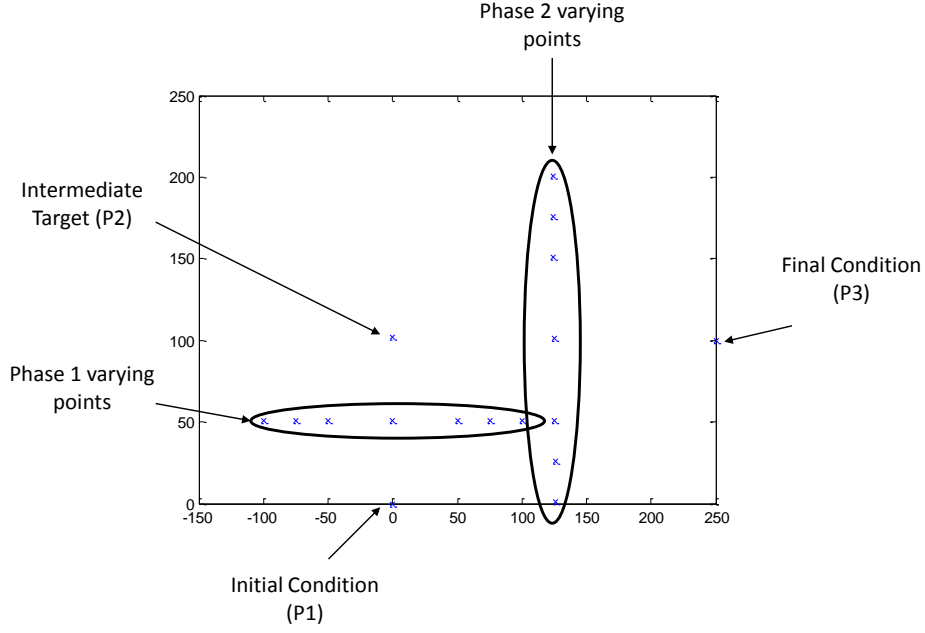
#### 8.4 Summary

The DoD is moving forward to produce appropriate autonomy in systems that work effectively and synergistically with manned counterparts. This research has demonstrated a methodology that provides optimal, rapid solutions to the loyal wingman optimal control problem *as defined herein* with an assumed level of autonomy. Regardless of which method is ultimately chosen, the methodology herein is an optimal solution that can aid in evaluating other dynamic route-planning algorithms. The body of work presented here is a foundation upon which to continue integrating the manned-unmanned team.

## Appendix A. 2D Deterministic Method to Produce Seeds

For an optimal control problem, a particle of a PSO algorithm is a vector of discrete-time control inputs that produce an optimal mission path with respect to the identified cost function. The loyal wingman PSO was seeded using a deterministic algorithm with a goal of meeting the following criteria: meet target and endpoint constraints, represent a broad range of possible trajectories to aid the PSO's stochastic search, and achievable in a computationally efficient manner.

The first step is to produce a set of possible two-dimensional curves using a spline interpolation. Initial, intermediate target, and final conditions, denoted  $p_1$ ,  $p_2$ , and  $p_3$ , are criteria established in the problem scenarios and provide a means to ensure the spline fit data meets constraint criteria. For purposes of this discussion, consider phase 1 between  $p_1$  and  $p_2$ , and phase 2 as between  $p_2$  and  $p_3$ . In order to allow for a broad range of trajectories, intermediate points are chosen in each of the phases. The Euclidean distance connecting  $p_1$  to  $p_2$  and  $p_2$  and  $p_3$  is computed as  $d_1$  and  $d_2$ , respectively. Beginning with phase 1, a perpendicular bisector  $L_1$ , the length of  $d_1$  is constructed at the midpoint between  $p_1$  and  $p_2$ . Points are now chosen on this perpendicular bisector to add curvature to the splines. If for example,  $\alpha$  points are chosen on  $L_1$ , then there are  $\alpha$  curves which can now connect  $p_1$  to  $p_2$ . Using the same method, sample points are also chosen in phase 2, such that if  $\beta$  points are chosen there are  $\beta$  curves which can now connect  $p_2$  to  $p_3$ . When the two phases are combined there are now  $M = \alpha * \beta$  possible splines that can be constructed that meet initial, intermediate target, and endpoint location criteria. The constructed points can be seen visually in Figure A.1. After the points through which a spline interpolation is desired have been identified, curves are parameterized using Eugene Lee's centripetal scheme, [148] through a convenient MATLAB function *cscvn*, which is the accumulated square root of the chord length. The MATLAB function *spline* can



**Figure A.1. Data Points Used to Fit Spline**

then be used which allows for specification of derivative conditions at the boundaries. In the case of the loyal wingman, the derivative boundary condition is heading,  $\psi$ , computed as  $\frac{dy}{dx}$ . Any initial heading can be supplied in the construction of the spline by specifying the initial condition boundary as  $y = \sin(\psi)$ ,  $x = \cos(\psi)$ . Specification of the boundary condition is very important in the loyal wingman PSO because the algorithm will simulate the controls with an identified initial heading. If the spline fit does not return an initial  $\frac{dy}{dx}$  equivalent to the initial condition specified by the loyal wingman problem, then the simulation will not produce a trajectory that meets constraints. Choosing  $\alpha = \beta = 7$  and  $\psi_1 = \frac{\pi}{2}$ ,  $\psi_{fc} = 0$ , the returned 49 spline fit curves can be seen in Figure A.2. No consideration is given to avoiding threats when producing these trajectories. This allows for the deterministic production of seeds as a general algorithm that can be used as the loyal wingman optimal control

problem scenarios are varied. At this point, a specified set of points (in this case 100

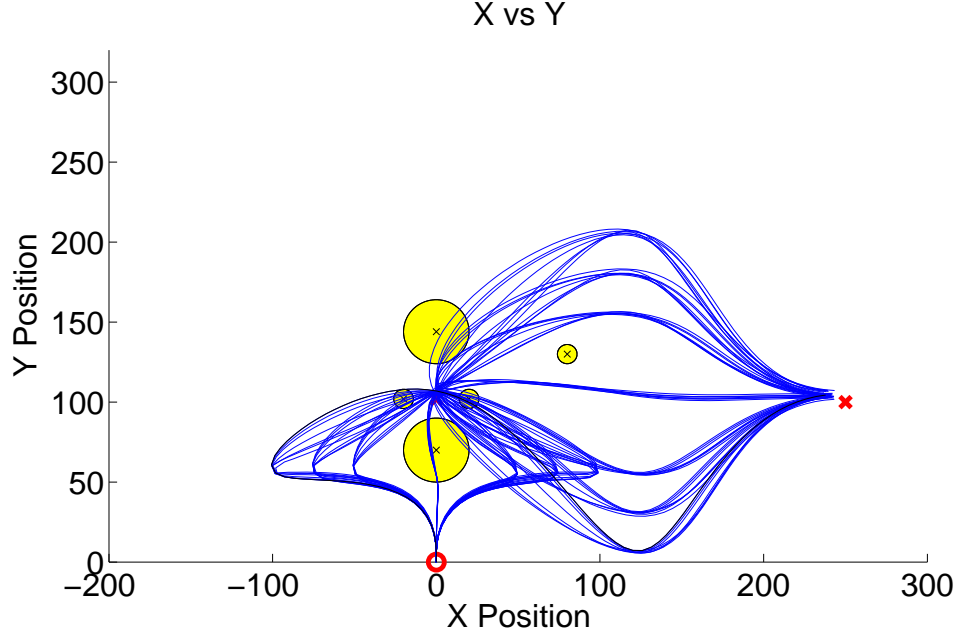


Figure A.2. 49 Trajectories Representing 49 Control Vector ‘Particles’

evenly spaced points for each spline) have been identified through a parameterized spline interpolation, but the goal is to achieve a set of controls as particles in the PSO algorithm. This is achieved by using the data output from the spline function to derive heading and heading rate. Before this can be done, the data must be re-parameterized to time. The arc length between each data point is computed using Equation A.1

$$arc(i) = \sqrt{1 + \left( \frac{y(i+1) - y(i)}{x(i+1) - x(i)} \right)^2} (x(i+1) - x(i)) \quad (A.1)$$

then divided by the loyal wingman’s constant velocity to achieve a time parameterization of the data points. The data points are then re-discretized using a spline interpolation to evenly spaced time units. There are now 49 trajectories composed of evenly spaced data points parameterized to time. From this, heading can be derived

at each time step  $i$  using

$$\psi(i) = \tan^{-1} \frac{y(i+1) - y(i)}{x(i+1) - x(i)}, \quad (\text{A.2})$$

and finally the heading rate control can be derived at each time step through the discrete equation of motion

$$u(i) = \frac{\psi(i+1) - \psi(i)}{\Delta t}. \quad (\text{A.3})$$

The desired particles, heading rate control vectors, have now been achieved. However, the time step for each trajectory is slightly different and in order to fit into the loyal wingman's PSO architecture, the time steps must be equivalent for all trajectories. A common vector length, chosen based on the longest trajectory and even time step is chosen to which all control vectors are re-fit. In cases where the common vector length is beyond what is needed to simulate the trajectory, the remaining elements are filled with 0. This allows the length of the trajectory to grow and shrink to the tune of the PSO algorithm's iterations.

## Appendix B. 3D Deterministic Method to Produce Seeds

Appendix A described a deterministic method for producing seeds for a two-dimensional model. The purpose of this index is to update a portion of Appendix A for the production of seeds in a three-dimensional model.

The first step is to produce a set of possible three-dimensional trajectories with Euclidean state space  $(x, y, z)$  data alone, using a spline interpolation. Referring to Figure A.1, initial, intermediate target, and final conditions, denoted  $p_1$ ,  $p_2$ , and  $p_3$ , respectively, are criteria established in the problem scenarios and provide a means to ensure the spline fit data meets constraint criteria. For purposes of this discussion, consider phase 1 between  $p_1$  and  $p_2$ , and phase 2 as between  $p_2$  and  $p_3$ . In order to allow for a broad range of trajectories, intermediate points are chosen in each of the phases. Considering only phase one, the Euclidean distance connecting  $p_1$  to  $p_2$  is computed as,

$$d_1 = \sqrt{(dx^2 + dy^2 + dz^2)}, \quad (\text{B.1})$$

where  $dx = x_1 - x_0$ ,  $dy = y_1 - y_0$ , and  $dz = z_1 - z_0$ .

The elevation and azimuth in global coordinates are computed as

$$\psi = \tan^{-1} \frac{dy}{dx} \quad (\text{B.2a})$$

$$\gamma = \sin^{-1} \frac{dz}{d_1} \quad (\text{B.2b})$$

In order to construct a set of candidate points, consider a set of equally distributed distances,  $\vec{r}_{1xL} \in [0, d_1]$ , representing the radii of various circles, and  $\vec{\psi}_{1xM}$ , evenly distributed  $\in [0, 2\pi]$ , representing an angular position on each of the circles. Algorithm 1 may then be used to construct a set of candidate phase one intermediate points through which a set of candidate splines may be fit.

Figure B.1 indicates the production of intermediate spline points. The top subplot

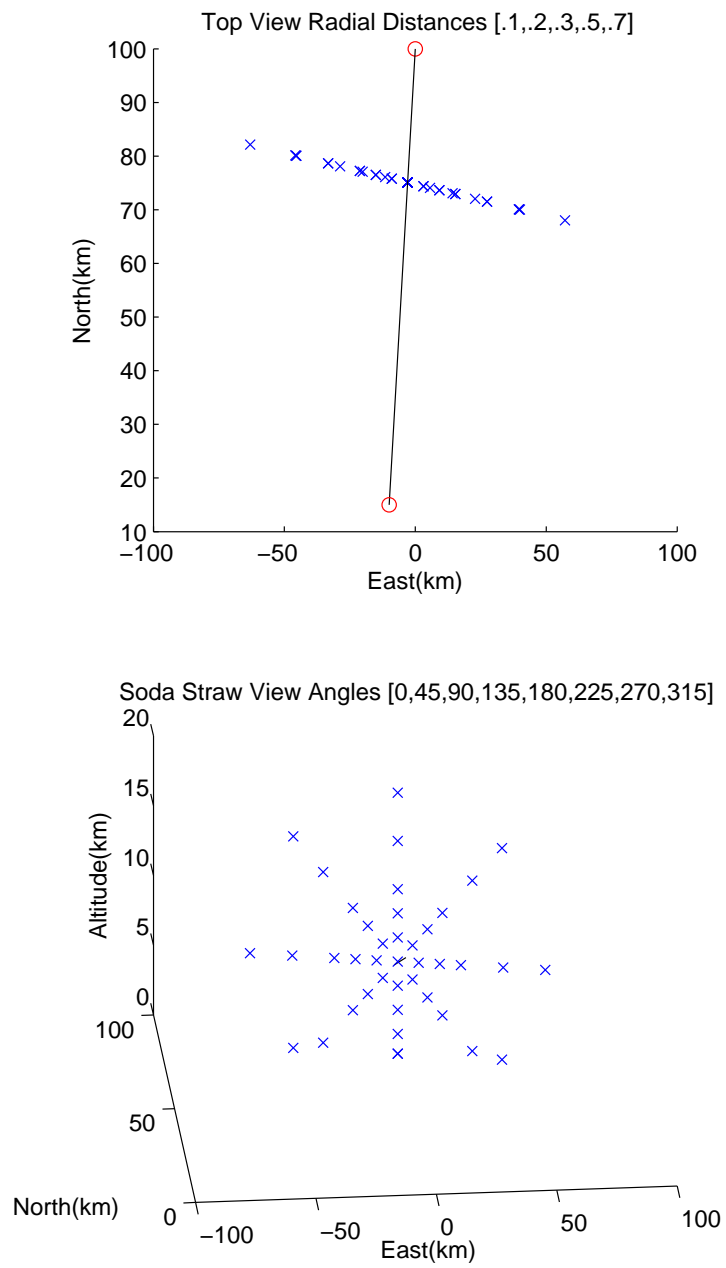


Figure B.1. Points Generated in 3D for Spline Interpolation

---

**Algorithm 1** Algorithm to Produce Candidate Points Through Which to Fit a Spline

---

```
1: for  $i = 1 : L$  do
2:   for  $j = 1 : M$  do
3:     for  $k = 1 : L * M$  do
4:        $x(k) = x_0 + r(i) \cdot \cos(-\gamma + \beta(j)) \cdot \cos(-\psi + \frac{\pi}{2})$ 
5:        $y(k) = y_0 + r(i) \cdot \cos(-\gamma + \beta(j)) \cdot \sin(-\psi + \frac{\pi}{2})$ 
6:        $z(k) = z_0 + r(i) \cdot \sin(-\gamma + \beta(j))$ 
7:     end for
8:   end for
9: end for
```

---

is a ‘top-view’ orientation, intended to highlight the radial distance for generating points. The bottom subplot is orientated in a view through a soda straw that connects  $p_1$  and  $p_2$  to indicate angular positions in concentric circles.

Finally, the derivative boundary conditions are used in the spline fit.

$$\begin{bmatrix} dx_0 \\ dy_0 \\ dz_0 \end{bmatrix} = V \begin{bmatrix} \cos(\gamma_0 \cdot \cos(\psi_0)) \\ \sin(\gamma_0 \cdot \cos(\psi_0)) \\ \sin(\gamma_0) \end{bmatrix} \quad (\text{B.3})$$

Once the points are created, the process continues as was indicated in Appendix A.

## Appendix C. Experiment to Choose PSO Parameters

The PSO algorithm, Equation C.1, contains various parameters such as social weightings,  $b_1, b_2$  [97] and a constriction factor,  $K$  [100],

$$\vec{v}(k+1) = K[\vec{v}(k) + b_1\vec{r}_1 \otimes (\vec{u}(k) - \vec{L}) + b_2\vec{r}_2 \otimes (\vec{u}(k) - \vec{G})] \quad (\text{C.1a})$$

$$\vec{u}(k+1) = \vec{u}(k) + \vec{v}(k+1). \quad (\text{C.1b})$$

Chapter IV highlighted convergence tendencies associated with the choice of parameters. An experiment was performed to determine the most appropriate value for the constriction factor as well as the choice of  $b_1$  and  $b_2$  for use in the loyal wingman optimal control problem.

The value  $\phi$  was varied from 4.1 to 7 along with the associated constriction factor as determined from Equation 4.2. The PSO algorithm was run 10 times for each value of  $\phi$  and the average cost, calculated using Gaussian quadrature, and time to converge were recorded. The results can be seen in Table C.1. The best cost in this experiment was with a  $\phi$  of 4.2 and constriction factor of .6417.  $\phi$  is a sum of the two

**Table C.1. Average Cost of Various Constriction Factors**

$\phi$	K	Cost	Iterations
7.0	.2087	2.4670	102
6.2	.2534	2.4587	124
5.8	.2845	2.4371	140
5.0	.3820	2.4130	180
4.2	.6417	2.3194	452
4.1	.7298	2.3402	454

social weighting factors so a separate experiment was run, where  $\phi$  and K are fixed at 4.2 and .6417, respectively, but the individual social components,  $b_1$  and  $b_2$ , are varied. Each scenario is run 10 times and the average cost is calculated and captured in Table C.2. The best cost was found when  $b_1$  was at 1 or less. This result is because

a high local weighting causes the search to stay in the local area, never moving its search toward the global best. When global best,  $b_2$  is weighted high, the particles search outside their local area in a movement toward the globally best particle.

**Table C.2. Average Cost of Various Social Weighting Factors**

$b_1$	$b_2$	Cost
4.0	0.2	2.4114
3.5	0.7	2.3468
3.0	1.2	2.3420
2.5	1.7	2.3127
1.5	2.7	2.2817
1.0	3.2	2.2688
0.2	4.0	2.2751
0.1	4.1	2.2715

## Appendix D. Dynamic, Stochastic Threat Size and Shape

Chapter VI established two parameters to form the size and shape of a moving, stochastic threat. The purpose of this appendix is to discuss the rationale behind the choice of Equations 6.5 and 6.6.

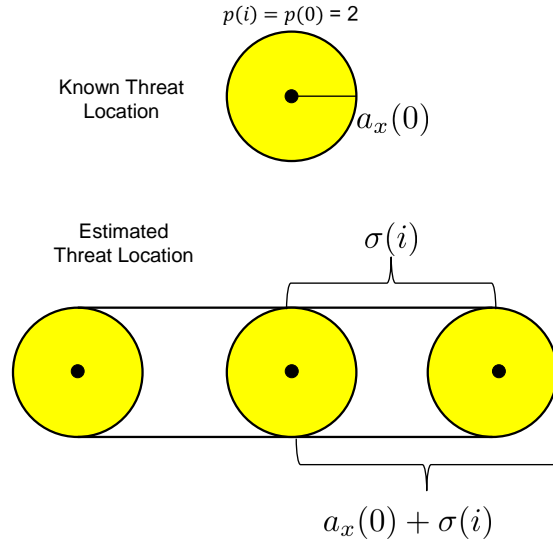
Referring to Figure D.1, the first circle represents a known threat location with known shaping parameter,  $p(i) = p(0) = 2$  to form an ellipse whose semi-minor x-axis,  $a_x(0)$ , is aligned with the road. When the threat is moving and is non-deterministic, this work estimates the location with a standard deviation using a Kalman filter. Referring to the bottom set of circles in Figure D.1, when estimating the threat location, it is not known if the threat is at the filter estimated position (center circle) or if it is a standard deviation away (two end circles) or somewhere in between. This work proposes avoidance of the entire region of a threat's possible location at each time-step. Two parameters are used to model the stochastic threat avoidance region. The first is the size parameter

$$a(i) = a_x(0) + \sigma(i) \tag{D.1}$$

$\forall i = 1 : N$  timesteps, which extends the axis along the road by the standard deviation  $\sigma(i)$  at each timestep. The shaping parameter is chosen to keep constant the axis distance perpendicular to the road,  $a_y$ , and vertical,  $a_z$ , while extending the axis along the road as a cylindrical shape according to

$$p(i) = p(0) + \frac{\sigma(i)}{a(0)}. \tag{D.2}$$

$\forall i = 1 : N$  timesteps. When the standard deviation is small in relation to the original length of the axis along the road,  $a_x(0)$ , the shaping parameter  $p \cong 2$  and the superquadric takes on the appearance of an ellipse. However, as the standard



**Figure D.1. Known and Unknown Threat Locations**

deviation increases, the shaping parameter forms a cylindrical shape whose curvature flattens along the distance-increasing axis of the road as well as the bases of the cylinder on each end. Results in Chapter VI show the changing size and shape of the superquadric as the threat stochastics are propagated.

## Appendix E. Distance to Re-Plan Formulation

Chapter VII provided set of equations to determine a time to replan,  $t_r$ , based on assumed threat speed and stochastics as well as the relative distance and angle of approach of the loyal wingman to the threat. This appendix demonstrates through simulation, the effect Equations 7.1, 7.4, 7.5, and 7.6 have on determining a time to re-plan. Referring to the first subplot of Figure E.1, a grid layout shows the loyal wingman and threat time-dependent trajectories.

The location of the loyal wingman and the threat are indicated at time-steps T1 through T4. T1 is where the simulation begins. Between T1 and T2, the center subplot indicates an angle of approach that is oblique, close to head-on and does not change. The value of  $\xi$  increases slowly due to the moving threat, while the bottom subplot indicates the distance between the two bodies is slowly increasing.

Between time-steps T2 and T3, the loyal wingman maneuvers and at T3, the angle of approach to the threat vector is perpendicular. This can be seen in the center subplot where the value of the angle of approach and  $\xi$  drop to zero. This angle of approach is only temporary, though and between T3 and T4, the angle of approach steadies. The bottom subplot indicates a slow decrease in the distance between the two bodies, causing  $\xi$  and  $\rho$  to increase. This continues until Equation 7.2 is satisfied at T4 and a re-plan occurs.

Referring to Figure E.2, a new trajectory has been produced at T4 and  $\xi$  is reset to 0 because the distance the threat has traveled is reset to 0. Between T4 and T5, the distance between the two bodies decreases, causing  $\xi$  and  $\rho$  to increase. This time, however, Equation 7.2 is not satisfied and at T5, the loyal wingman and threat begin to move away from each other. Because the loyal wingman is traveling at a speed that is much greater than the threat,  $H$  and  $\rho$  diverge and Equation 7.2 is never satisfied, i.e., there is no re-plan.

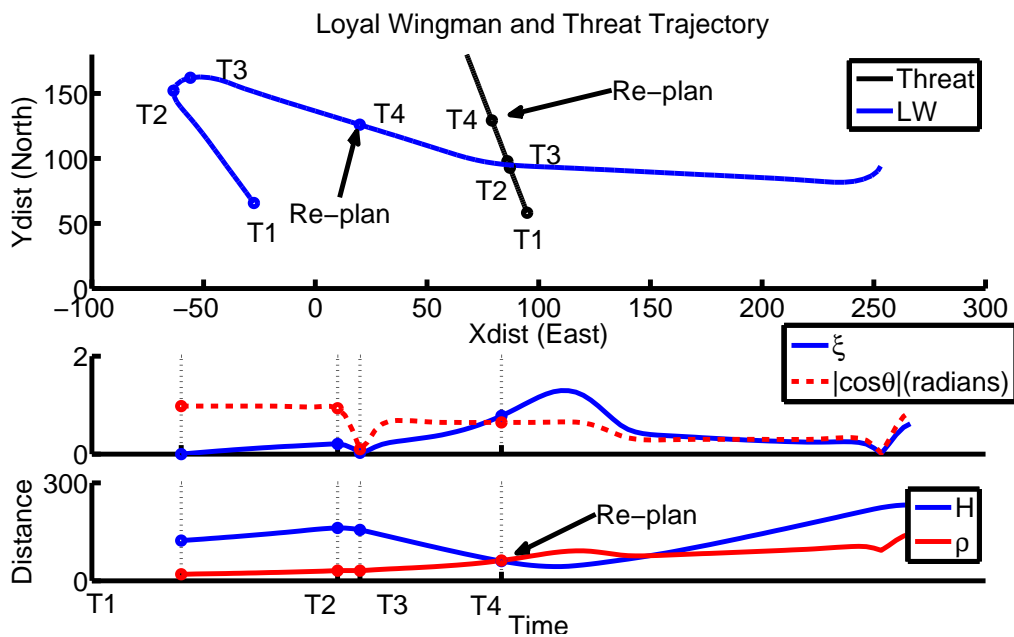


Figure E.1. Trajectory 1: Effect of  $\rho$ ,  $\xi$ , and  $H$

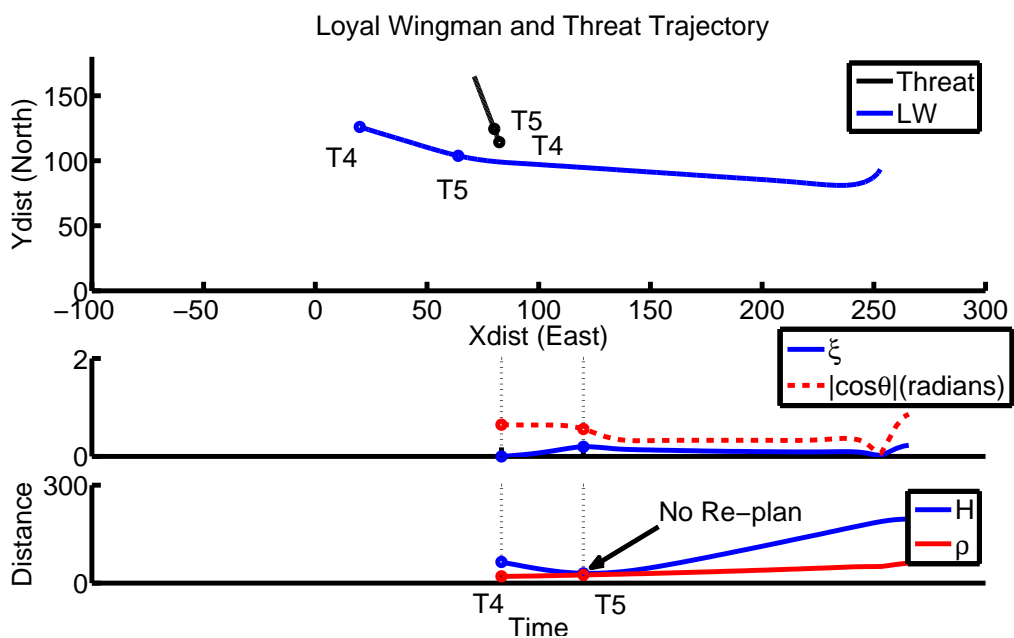


Figure E.2. Trajectory 2: Effect of  $\rho$ ,  $\xi$ , and  $H$

## Bibliography

1. Jr. Winnefeld, James A and Frank Kendall. 2013-2036 unmanned systems integrated roadmap. US Department of Defense, 2013.
2. James A Winnefeld and Frank Kendall. Unmanned systems integrated roadmap fy 2011-2036, 2011. “[http://www.dtic.mil/ndia/2011MCSC/Thompson\\_UnmannedSystems.pdf](http://www.dtic.mil/ndia/2011MCSC/Thompson_UnmannedSystems.pdf)”.
3. Deborah Lee James and Mark A. Welsh, III. United states air force rpa vector: Vision and enabling concepts 2013-2038, 2014. “[http://www.defenseinnovationmarketplace.mil/resources/USAF-RPA\\_VectorVisionEnablingConcepts2013-2038\\_ForPublicRelease.pdf](http://www.defenseinnovationmarketplace.mil/resources/USAF-RPA_VectorVisionEnablingConcepts2013-2038_ForPublicRelease.pdf)”.
4. WJA Dahm. Technology horizons a vision for air force science & technology during 2010-2030, 2010. “[www.dtic.mil/dtic/tr/fulltext/u2/a525912.pdf](http://www.dtic.mil/dtic/tr/fulltext/u2/a525912.pdf)”.
5. Air force research laboratory autonomy science and technology strategy, 2013.
6. ML Cummings and D Morales. Uavs as tactical wingmen: Control methods and pilots perceptions. *Unmanned Systems*, 23(1):25–27, 2005.
7. Benjamin R Carter. Time-optimization of high performance combat maneuvers. Technical report, Naval Postgraduate School, 2005.
8. Nathan Smith. *Optimal Collision Avoidance Trajectories for Unmanned/Remotely Piloted Aircraft*. PhD thesis, Air Force Institute of Technology (AU), Wright Patterson AFB, OH, 2014. AFIT-ENY-DS-14-D-34.
9. Damien Galzi and Yuri Shtessel. Uav formations control using high order sliding modes. In *IEEE American Control Conference*, 2006. doi: 10.1109/ACC.2006.1657386.
10. Nathan Harl and SN Balakrishnan. Coordinated rendezvous of unmanned air vehicles to a formation: A sliding mode approach. In *AIAA Guidance, Navigation and Control Conference*, 2008.
11. Srdjan S Stankovic, Milorad J Stanojevic, and Dragoslav D Siljak. Decentralized overlapping control of a platoon of vehicles. *IEEE Transactions on Control Systems Technology*, 8(5):816–832, 2000. doi: 10.1109/87.865854.
12. Sriram Venkataramanan and Atila Dogan. A multi-uav simulation for formation reconfiguration. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit, Providence, RI*, 2004.
13. Sriram Venkataramanan and Atila Dogan. Nonlinear control for reconfiguration of uav formation. In *AIAA Guidance, Navigation, and Control Conference, Proceedings of the*, 2003.

14. Alan H Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.
15. Bruce T Clough. Unmanned aerial vehicles: Autonomous control challenges, a researcher’s perspective. *Journal of Aerospace Computing, Information, and Communication*, 2(8):327–347, 2005. doi: 10.2514/1.5588.
16. Ji Hyun Yang, Marek Kapolka, and Timothy H Chung. Autonomy balancing in a manned-unmanned teaming (mut) swarm attack. In *Robot Intelligence Technology and Applications 2012*, pages 561–569. Springer, 2013. doi: 10.1007/978-3-642-37374-9-54.
17. Stefan Gangl, Benjamin Lettl, and Axel Schulte. Management of multiple unmanned combat aerial vvehicles from a single-seat fighter cockpit in manned-unmanned fighter missions. *AIAA Infotech@ Aerospace*, pages 1–18, 2013. doi:10.2514/6.2013-4899.
18. Fabian Schmitt and Axel Schulte. Mixed-initiative interaction in manned-unmanned-teaming mission planning: Design and evaluation of a prototype. In *AIAA Infotech@ Aerospace Conference*, 2015. doi:10.2514/6.2015-0114.
19. Steve Jameson, Jerry Franke, Robert Szczerba, and Sandy Stockdale. Collaborative autonomy for manned/unmanned teams. In *ANNUAL FORUM PROCEEDINGS-AMERICAN HELICOPTER SOCIETY*, volume 61, page 1673. AMERICAN HELICOPTER SOCIETY, INC, 2005.
20. Peter Svenmarck, Patrik Lif, Hans Jander, and Jonathan Borgvall. *Studies of Manned-Unmanned Teaming Using Cognitive Systems Engineering: An Interim Report*. Totalförsvarets forskningsinstitut (FOI), 2005. ISSN: 1650-1942.
21. Lt Col Van Riper, Steven G. Apache manned-unmanned teaming capability. *Army Magazine*, 64.
22. David B Durbin and Jamison S Hicks. Ah-64d apache longbow aircrew workload assessment for unmanned aerial system (uas) employment. Technical report, DTIC Document, 2009. dtic accession #:ADA494123.
23. Richard C. Young. *X-47B Unmanned Combat Air System (UCAS)*, November 15, 2012, accessed October 30, 2014.
24. US Air Force. Us air force manual 10/100, airman’s manual, 1999.
25. James S McGrew, Jonathon P How, Brian Williams, and Nicholas Roy. Air-combat strategy using approximate dynamic programming. *Journal of Guidance, Control, and Dynamics*, 33(5):1641–1654, 2010.
26. Todd A Murphey. Filling the airpower vacuum. Technical report, DTIC Document, 2013.

27. *Two Global Hawk Unmanned Aircraft Fly in Close Formation, Move AHR Program Closer to Autonomous Aerial Refueling*, accessed October 30, 2014.
28. Stephen Waydo, John Hauser, Robert Bailey, Eric Klavins, and Richard M Murray. Uav as a reliable wingman: A flight demonstration. *Control Systems Technology, IEEE Transactions on*, 15(4):680–688, 2007. dtic accesssion #: ADA464849.
29. Richard D Garcia, Laura Barnes, and MaryAnne Fields. Unmanned aircraft systems as wingmen. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, 9(1):5–15, 2012. doi: 10.1177/1548512910391217.
30. Fabrizio Giulietti, Lorenzo Pollini, and Mario Innocenti. Autonomous formation flight. *Control Systems, IEEE*, 20(6):34–44, 2000.
31. Austin L Smith. Proportional navigation with adaptive terminal guidance for aircraft rendezvous. *Journal of Guidance, Control, and Dynamics*, 31(6):1832–1836, 2008. doi: 10.2514/1.33535.
32. Stephen A Murtaugh and Harry E Criel. Fundamentals of proportional navigation. *Spectrum, IEEE*, 3(12):75–85, 1966.
33. Sanghyuk Park. *Avionics and Control System Development for Mid-Air Rendezvous of Two Unmanned Aerial Vehicles*. PhD thesis, Massachusetts Institute of Technology, 2003.
34. Min-Jea Tahk, Chang-Su Park, and Chang-Kyung Ryoo. Line-of-sight guidance laws for formation flight. *Journal of Guidance, Control, and Dynamics*, 28(4):708–716, 2005. doi: 10.2514/1.9605.
35. K David Young, Vadim I Utkin, and Umit Ozguner. A control engineer’s guide to sliding mode control. *Control Systems Technology, IEEE Transactions on*, 7(3):328–342, 1999.
36. Heather M Stickney. Performance characterization, development, and application of artificial potential function guidance methods. Technical report, DTIC Document, 2014.
37. Laura E Barnes, Mary-Anne Fields, and Kimon P Valavanis. Swarm formation control utilizing elliptical surfaces and limiting functions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(6):1434–1445, 2009.
38. Tobias Paul, Thomas R Krogstad, and Jan Tommy Gravdahl. Modelling of uav formation flight using 3d potential field. *Simulation Modelling Practice and Theory*, 16(9):1453–1462, 2008.

39. Herbert G Tanner, Ali Jadbabaie, and George J Pappas. Stable flocking of mobile agents, part i: Fixed topology. In *Decision and Control, Proceedings of the 42nd IEEE Conference on*, volume 2, pages 2010–2015. IEEE, 2003.
40. O Ilaya, C Bill, and M Evans. Control design for unmanned aerial vehicle swarming. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 222(4):549–567, 2008.
41. Herbert G Tanner, Ali Jadbabaie, and George J Pappas. Stable flocking of mobile agents part ii: Dynamic topology. In *Decision and Control, Proceedings of the 42nd IEEE Conference on*, volume 2, pages 2016–2021. IEEE, 2003.
42. Michael A Kovacina, Daniel Palmer, Guang Yang, and Ravi Vaidyanathan. Multi-agent control algorithms for chemical cloud detection and mapping using unmanned air vehicles. In *International Conference on Intelligent Robots and Systems*, volume 3, pages 2782–2788. IEEE, 2002. doi: 10.1109/IRDS.2002.1041691.
43. Felipe Cucker and Steve Smale. Emergent behavior in flocks. *IEEE Transactions on Automatic Control*, 52(5):852–862, 2007. doi: 10.1109/TAC.2007.895842.
44. Derek W Ebdon. Model predictive control of aerospace systems. Technical report, DTIC Document, 1996.
45. Zhao Weihua and Tiauw Hiong Go. Robust cooperative leader-follower formation flight control. In *Control Automation Robotics & Vision (ICARCV), Proceedings of the 2010 11th International Conference on*, pages 275–280. IEEE, 2010. 10.1109/ICARCV.2010.5707246.
46. Xiaohua Wang, Vivek Yadav, and SN Balakrishnan. Cooperative uav formation flying with obstacle/collision avoidance. *Control Systems Technology, IEEE Transactions on*, 15(4):672–679, 2007.
47. Stephen Grossberg. Nonlinear neural networks: Principles, mechanisms, and architectures. *Neural networks*, 1(1):17–61, 1988.
48. Zhou Chao, Lei Ming, Zhou Shaolei, and Zhang Wenguang. Collision-free uav formation flight control based on nonlinear mpc. In *Electronics, Communications and Control (ICECC), 2011 International Conference on*, pages 1951–1956. IEEE, 2011.
49. Jovan D Boskovic, Sai-Ming Li, and Raman K Mehra. Semi-globally stable formation flight control design in three dimensions. In *Decision and Control, Proceedings of the 40th IEEE Conference on*, volume 2, pages 1059–1064. IEEE, 2001.

50. Jovan D Boskovic, Sai-Ming Li, and Raman K Mehra. Formation flight control design in the presence of unknown leader commands. In *American Control Conference, Proceedings of the 2002*, volume 4, pages 2854–2859. IEEE, 2002.
51. Sai-Ming Li, Jovan D Boskovic, and Raman K Mehra. Globally stable automatic formation flight control in two dimensions. In *Proceedings of AIAA Guidance, Navigation and Control Conference, Montreal, Canada, 6-9 August*, 2001.
52. Zhao Weihua and Tiau Hiong Go. Robust decentralized formation flight control. *International Journal of Aerospace Engineering*, 2011, 2011.
53. Keck Voon Ling, Jan Maciejowski, Arthur Richards, and Bing Fang Wu. Multiplexed model predictive control. *Automatica*, 48(2):396–401, 2012.
54. Hai-bin Duan, Guan-jun Ma, and De-lin Luo. Optimal formation reconfiguration control of multiple ucavs using improved particle swarm optimization. *Journal of Bionic Engineering*, 5(4):340–347, 2008. doi:10.1016/S1672-6529(08)60179-1.
55. Guangfu Ma, Haibin Huang, and Yufei Zhuang. Time optimal trajectory planning for reconfiguration of satellite formation with collision avoidance. In *Control and Automation (ICCA), 2010 8th IEEE International Conference on*, pages 476–479. IEEE, 2010.
56. Matthew A Russell and Gary B Lamont. A genetic algorithm for unmanned aerial vehicle routing. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, pages 1523–1530. ACM, 2005.
57. Haibin Duan, Qinan Luo, Yuhui Shi, and Guan-jun Ma. Hybrid particle swarm optimization and genetic algorithm for multi-uav formation reconfiguration. *IEEE Computational Intelligence Magazine*, 8(3):16–27, 2013. doi:10.1109/MCI.2013.2264577.
58. Claude Sammut and Geoffrey I Webb. *Encyclopedia of Machine Learning*. Springer Science & Business Media, 2011.
59. Duvsan Teodorovic and Mauro Dell’Orco. Bee colony optimization—a cooperative learning approach to complex transportation problems. In *Advanced OR and AI Methods in Transportation, Proceedings of 16th Mini-EURO Conference*, pages 51–60. Poznan Publishing House of the Polish Operational and System Research, 2005.
60. Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. *ACM Siggraph Computer Graphics*, 21(4):25–34, 1987. doi:10.1145/37401.37406.
61. Jacob L Lambach. Integrating uas flocking operations with formation drag reduction. Master’s thesis, Air Force Institute of Technology (AU),

Wright-Patterson AFB OH, 2014. AFIT-ENV-14-M-01DL, dtic accession #: ADA601735.

62. Jennifer N Kaiser. Effects of dynamically weighting autonomous rules in an unmanned aircraft system (uas) flocking model. Master's thesis, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 2014. AFIT-ENV-T-14-S-06, dtic accession #: ADA608924.
63. CA Rabbath, C-Y Su, and Antonios Tsourdos. Guest editorial introduction to the special issue on multivehicle systems cooperative control with application. *Control Systems Technology, IEEE Transactions on*, 15(4):599–600, 2007.
64. Phillip Chandler, Steven Rasmussen, and Meir Pachter. Uav cooperative path planning. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1255–1265, 2000.
65. Timothy W McLain, Phillip R Chandler, Steven Rasmussen, and Meir Pachter. Cooperative control of uav rendezvous. In *American Control Conference, Proceedings of the 2001*, volume 3, pages 2309–2314. IEEE, 2001.
66. Tal Shima, Steve Rasmussen, and Dave Gross. Assigning micro uavs to task tours in an urban terrain. *Control Systems Technology, IEEE Transactions on*, 15(4):601–612, 2007.
67. Allison Ryan, John Tisdale, Mark Godwin, Daniel Coatta, David Nguyen, Stephen Spry, Raja Sengupta, and J Karl Hedrick. Decentralized control of unmanned aerial vehicle collaborative sensing missions. In *American Control Conference, Proceedings of the 2007*, pages 4672–4677. IEEE, 2007.
68. Mehdi Alighanbari and Jonathan P How. Robust decentralized task assignment for cooperative uavs. In *Guidance, Navigation, and Control Conference and Exhibit, Proceedings of the AIAA*, pages 21–24, 2006.
69. Mehdi Alighanbari and Jonathan P How. Decentralized task assignment for unmanned aerial vehicles. In *Decision and Control, 44th IEEE Conference on*, pages 5668–5673. IEEE, 2005. 10.1109/CDC.2005.1583066.
70. Tomonari Furukawa, Hugh F Durrant-Whyte, Frédéric Bourgault, and Gamini Dissanayake. Time-optimal coordinated control of the relative formation of multiple vehicles. In *Computational Intelligence in Robotics and Automation, Proceedings 2003 IEEE International Symposium on*, volume 1, pages 259–264. IEEE, 2003. DOI: 10.1109/CIRA.2003.1222099.
71. Donald E Kirk. *Optimal Control Theory: An Introduction*. Courier Corporation, 2012.

72. Timothy R Jorris and Richard G Cobb. Three-dimensional trajectory optimization satisfying waypoint and no-fly zone constraints. *Journal of Guidance, Control, and Dynamics*, 32(2):551–572, 2009. doi: 10.2514/1.37030.
73. Tadeusz J Masternak. *Multi-Objective Trajectory Optimization of a Hypersonic Reconnaissance Vehicle with Temperature Constraints*. PhD thesis, Air Force Institute of Technology (AU), Wright-Patterson AFB, OH, 2014. AFIT-ENV-DS-14-D-21.
74. Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12(4):979–1006, 2002. doi:10.1137/S1052623499350013.
75. Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006. doi:10.1007/s10107-004-0559-y.
76. David Benson. *A Gauss Pseudospectral Transcription for Optimal Control*. PhD thesis, Massachusetts Institute of Technology, 2005. URL:<http://hdl.handle.net/1721.1/28919>.
77. Geoffrey Huntington. *Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control*. PhD thesis, Massachusetts Institute of Technology, 2007.
78. Michael A Patterson and Anil V Rao. Version 2.0: A general-purpose matlab software for solving multiple-phase optimal control problems.
79. MATLAB. *version 7.14.0 (R2014a)*. The MathWorks Inc., Natick, Massachusetts, 2014.
80. Nidal M Jodeh and Richard G Cobb. Optimal airborne trajectories for data collection from wireless sensor networks by direct collocation methods. In *Guidance, Navigation and Control Conference, Proceedings of the AIAA*, 2015.
81. Angela W Suplisson, Richard G Cobb, William P Baker, and David R Jacques. An optimal control approach to aircraft automatic ground collision avoidance.
82. Jorge De La Mata. Optimal strategies for selecting from multiple interception targets. Master’s thesis, University of Illinois at Urbana-Champaign, 2014.
83. Anil V Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528, 2009.
84. Chao Zhang, Ziyang Zhen, Daobo Wang, and Meng Li. Uav path planning method based on ant colony optimization. In *Control and Decision Conference (CCDC), 2010 Chinese*, pages 3790–3792. IEEE, 2010.

85. Rainer Storn and Kenneth Price. *Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces*, volume 3. ICSI Berkeley, 1995.
86. Scott Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5-6):975–986, 1984.
87. Bruce A Conway. A survey of methods available for the numerical optimization of continuous dynamic systems. *Journal of Optimization Theory and Applications*, 152(2):271–306, 2012. doi: 10.1007/s10957-011-9918-z.
88. Jacob A Englander, Bruce A Conway, and Trevor Williams. Automated mission planning via evolutionary algorithms. *Journal of Guidance, Control, and Dynamics*, 35(6):1878–1887, 2012. doi: 10.2514/1.54101.
89. Christian M Chilan and Bruce A Conway. Automated design of multiphase space missions using hybrid optimal control. *Journal of Guidance, Control, and Dynamics*, 36(5):1410–1424, 2013.
90. Daniel J Showalter and Jonathan T Black. Near-optimal geostationary transfer maneuvers with cooperative en-route inspection using hybrid optimal control. *Acta Astronautica*, 105(2):395–406, 2014. 10.1016/j.actaastro.2014.09.013.
91. Tamas Vinko and Dario Izzo. Global optimisation heuristics and test problems for preliminary spacecraft trajectory design. *ESA Advanced Concepts Team Technical Report*, 2008. website accessed 3 June 2016, <http://www.esa.int/gsp/ACT/doc/INF/pub/ACT-TNT-INF-2008-GOHTPPSTD.pdf>.
92. Hamidreza Modares and Mohammad-Bagher Naghibi Sistani. Solving non-linear optimal control problems using a hybrid ipso-sqp algorithm. *Engineering Applications of Artificial Intelligence*, 24(3):476–484, 2011. doi: 10.1016/j.engappai.2010.08.002.
93. Yufei Zhuang and Haibin Huang. Time-optimal trajectory planning for under-actuated spacecraft using a hybrid particle swarm optimization algorithm. *Acta Astronautica*, 94(2):690–698, 2014. doi:10.1016/j.actaastro.2013.06.023.
94. Sundaram Suresh, Hai-Jun Rong, and Narasimhan Sundararajan. Bio-inspired computing for launch vehicle design and trajectory optimization. In *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–8. IEEE, 2009. doi: 10.1109/CISDA.2009.5356548.
95. CH Yam, DD Lorenzo, and D Izzo. Low-thrust trajectory design as a constrained global optimization problem. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 225(11):1243–1251, 2011. doi: 10.1177/0954410011401686.

96. JA Englander, BA Conway, and T Williams. Automated interplanetary mission planning. In *AAS/AIAA Astrodynamics Specialist Conference, Minneapolis, MN*, 2012. doi: 10.2514/5.9781624102714.0669.0706.
97. Russ C Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *Sixth International Symposium on Micro Machine and Human Science, Proceedings of the*, volume 1, pages 39–43. New York, NY, 1995.
98. Maurice Clerc and James Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002. 10.1109/4235.985692.
99. Ioan Cristian Trelea. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters*, 85(6):317–325, 2003. 10.1016/S0020-0190(02)00447-7.
100. Maurice Clerc. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the Congress on Evolutionary Computation*, volume 3. IEEE, 1999. doi:10.1109/CEC.1999.785513.
101. Yonghe Lu, Minghui Liang, Zeyuan Ye, and Lichao Cao. Improved particle swarm optimization algorithm and its application in text feature selection. *Applied Soft Computing*, 35:629–636, 2015. doi:10.1016/j.asoc.2015.07.005.
102. Russ C Eberhart and Yuhui Shi. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 84–88. IEEE, 2000. doi:10.1109/CEC.2000.870279.
103. Rania Hassan, Babak Cohanime, Olivier De Weck, and Gerhard Venter. A comparison of particle swarm optimization and the genetic algorithm. In *Proceedings of the 1st AIAA Multidisciplinary Design Optimization Specialist Conference*, pages 18–21, 2005. doi: 10.2514/6.2005-1897.
104. Rega Rajendra, Dilip K Pratihari, et al. Particle swarm optimization algorithm vs genetic algorithm to develop integrated scheme for obtaining optimal mechanical structure and adaptive controller of a robot. *Intelligent Control and Automation*, 2(04):430, 2011. doi: 10.4236/ica.2011.24050.
105. Mauro Pontani and Bruce A Conway. Particle swarm optimization applied to space trajectories. *Journal of Guidance, Control, and Dynamics*, 33(5):1429–1441, 2010. doi: 10.2514/1.48475.
106. HB Duan and SQ Liu. Non-linear dual-mode receding horizon control for multiple unmanned air vehicles formation flight based on chaotic particle swarm optimisation. *Control Theory & Applications, IET*, 4(11):2565–2578, 2010. doi:10.1049/iet-cta.2009.0256.

107. Issam Mazhoud, Khaled Hadj-Hamou, Jean Bigeon, and Patrice Joyeux. Particle swarm optimization for solving engineering problems: A new constraint-handling mechanism. *Engineering Applications of Artificial Intelligence*, 26(4):1263–1273, 2013. doi:10.1016/j.engappai.2013.02.002.
108. Martin Saska, Martin Macaš, Libor Přeučil, and Lenka Lhotská. Robot path planning using particle swarm optimization of ferguson splines. In *IEEE Conference on Emerging Technologies and Factory Automation*, pages 833–839. IEEE, 2006. doi: 10.1109/ETFA.2006.355416.
109. Xiaohui Hu and Russell Eberhart. Solving constrained nonlinear optimization problems with particle swarm optimization. In *Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics*, volume 5, pages 203–206. Citeseer, 2002.
110. Peter Eberhart and Kai Sedlaczek. Using augmented lagrangian particle swarm optimization for constrained problems in engineering. In *Advanced Design of Mechanical Systems: From Analysis to Optimization*, pages 253–271. Springer, 2009. doi:10.1007/978-3-211-99461-0-12.
111. Haiyan Lu and Weiqi Chen. Self-adaptive velocity particle swarm optimization for solving constrained optimization problems. *Journal of Global Optimization*, 41(3):427–445, 2008. doi:10.1007/s10898-007-9255-9.
112. Chukiat Worasuchep. A particle swarm optimization with stagnation detection and dispersion. In *IEEE World Congress on Computational Intelligence*, pages 424–429. IEEE, 2008. doi:10.1109/CEC.2008/4630832.
113. Northrop Aerospace Systems. Design description document (ddd) for multi-sensor integrated conflict avoidance (musica), 2010. Contracting Agency: Air Force Research Laboratory, AFRL, RQQC, Wright-Patterson Air Force Base.
114. Zhao Weihua and Tiau Hiong Go. 3-d formulation of formation flight based on model predictive control with collision avoidance scheme. In *Proceedings of the 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010.
115. Branko Grünbaum, Victor Klee, Micha A Perles, and Geoffrey Colin Shephard. *Convex Polytopes*. Springer, 1967.
116. Jarurat Ousingsawat and Mark E Campbell. On-line estimation and path planning for multiple vehicles in an uncertain environment. *International Journal of Robust and Nonlinear Control*, 14(8):741–766, 2004. DOI: 10.1002/rnc.933.
117. Michael J Todd and E Alper Yildirim. On khachiyan’s algorithm for the computation of minimum-volume enclosing ellipsoids. *Discrete Applied Mathematics*, 155(13):1731–1744, 2007. doi:10.1016/j.dam.2007.02.013.

118. Nima Moshtagh. Minimum volume enclosing ellipsoid. *Convex Optimization*, 111:112, 2005.
119. Ken Shoemake. Animating rotation with quaternion curves. In *ACM SIGGRAPH computer graphics*, volume 19, pages 245–254. ACM, 1985. doi: 10.1145/325165.325242.
120. Ales Jaklic, Ales Leonardis, and Franc Solina. *Segmentation and Recovery of Superquadrics*, volume 20. Springer, 2000. ISBN-10: 0792366018.
121. Ryan D Gauntt. Aircraft course optimization tool using gpops matlab code. Master’s thesis, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, 2012. AFIT/GSE/ENV/12-M03, dtic accession # ADA557164.
122. X Rong Li and Vesselin P Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *Aerospace and Electronic Systems, IEEE Transactions on*, 39(4):1333–1364, 2003.
123. Robert A Singer. Estimating optimal tracking filter performance for manned maneuvering targets. *Aerospace and Electronic Systems, IEEE Transactions on*, (4):473–483, 1970.
124. David R Maroney<sup>1</sup> Robert H Bolling, Ravindra Athale, and Alan D Christiansen. Experimentally scoping the range of uas sense and avoid capability. 2007.
125. Robert H Chen, Arthur Gevorkian, Alex Fung, Won-Zon Chen, and Vincent Raska. Multi-sensor data integration for autonomous sense and avoid. In *AIAA Infotech at Aerospace Technical Conference*, 2011.
126. Randal W Beard, Derek Kingston, Morgan Quigley, Deryl Snyder, Reed Christiansen, Walt Johnson, Timothy McLain, and Michael Goodrich. Autonomous vehicle technologies for small fixed-wing uavs. *Journal of Aerospace Computing, Information, and Communication*, 2(1):92–108, 2005.
127. Jusuk Lee, Rosemary Huang, Andrew Vaughn, Xiao Xiao, J Karl Hedrick, Marco Zennaro, and Raja Sengupta. Strategies of path-planning for a uav to track a ground vehicle. In *Autonomous Intelligent Networks and Systems, Proceedings of the Second Annual Symposium on*, volume 2003, 2003.
128. David Benson. *A Gauss Pseudospectral Transcription for Optimal Control*. PhD thesis, Massachusetts Institute of Technology, 2005.
129. Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
130. Robert G. Brown and Patrick Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley, 2012.

131. Tal Shima, Steven J Rasmussen, and Phillip Chandler. Uav team decision and control using efficient collaborative estimation. *Journal of Dynamic Systems, Measurement, and Control*, 129(5):609–619, 2007. doi:10.1115/1.2764504.
132. Eric A Wan and Rudolph Van Der Merwe. The unscented kalman filter for non-linear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control, The IEEE Symposium on*, pages 153–158. IEEE, 2000.
133. Steven M Ross, Richard G Cobb, and William P Baker. Stochastic real-time optimal control for bearing-only trajectory planning. *International Journal of Micro Air Vehicles*, 6(1):1–28, 2014.
134. Jinwhan Kim, SS Vaddi, PK Menon, and EJ Ohlmeyer. Comparison between three spiraling ballistic missile state estimators. In *AIAA Guidance, Navigation and Control Conference, Proceedings of the*, 2008.
135. M Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *Signal Processing, IEEE Transactions on*, 50(2):174–188, 2002.
136. Roberto Conde, A Ollero, and JA Cobano. Method based on a particle filter for uav trajectory prediction under uncertainties. In *40th International Symposium of Robotics, Barcelona, Spain*, 2009.
137. Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and P-J Nordlund. Particle filters for positioning, navigation, and tracking. *Signal Processing, IEEE Transactions on*, 50(2):425–437, 2002.
138. Hui Yan, Fariba Fahroo, and I Michael Ross. Real-time computation of neighboring optimal control laws. *AIAA Paper*, 4657, 2002.
139. Zhesheng Jiang and Raul Ordonez. Trajectory generation on approach and landing for rlvs using motion primitives and neighboring optimal control. In *American Control Conference*, pages 1091–1096. IEEE, 2007.
140. William K McQuay. Distributed collaborative environments for 21st century modeling and simulation. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 164–173. International Society for Optics and Photonics, 2001.
141. Charles A Leavitt. Real-time in-flight planning. In *Aerospace and Electronics Conference, Proceedings of the IEEE 1996 National*, volume 1, pages 83–89. IEEE, 1996.
142. Clay J Humphreys, Richard G Cobb, David R Jacques, and Jonah A Reeger. Optimal mission paths for the uninhabited loyal wingman. *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2015. doi: 10.2514/6.2015-2792.

143. Anil V Rao, David A Benson, Christopher Darby, Michael A Patterson, Camila Francolin, Ilyssa Sanders, and Geoffrey T Huntington. Algorithm 902: Gpops, a matlab software for solving multiple-phase optimal control problems using the gauss pseudospectral method. *ACM Transactions on Mathematical Software (TOMS)*, 37(2):22, 2010.
144. Peter S Maybeck and George M Siouris. Stochastic models, estimation, and control, volume i. *IEEE Transactions on Systems, Man, and Cybernetics*, 5(10):282, 1980.
145. Michael A Patterson and Anil Rao. Exploiting sparsity in direct collocation pseudospectral methods for solving optimal control problems. *Journal of Spacecraft and Rockets*, 49(2):354–377, 2012. doi: 10.2514/1.A32071.
146. Michael A Patterson and Anil V Rao. Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software (TOMS)*, 41(1):1, 2014. doi: 10.1145/2558904.
147. Clay J Humphreys, Richard G Cobb, David R Jacques, and Jonah A Reeger. A hybrid technique to rapidly solve the intermediate-target optimal control problem. *Global Journal of Technology and Optimization*, submitted for publication, July 2016.
148. Eugene TY Lee. Choosing nodes in parametric curve interpolation. *Computer-Aided Design*, 21(6):363–370, 1989. doi: 10.1016/0010-4485(89)90003-1.

## Vita

Major Clay J Humphreys is a PhD student studying aircraft control systems and optimization in the Department of Aeronautics and Astronautics at the Air Force Institute of Technology (AFIT), WPAFB, OH. Major Humphreys was born and raised in Grand Prairie Texas, a suburb of Dallas and after graduating from Lamar High School in 1997, he attended Texas A & M University. He graduated in 2001 with a bachelor's degree in Aerospace Engineering and soon after commenced Air Force Officer Training School.

In 2001, after initial training, Clay was assigned as a modifications engineer to the 82d Aerial Targets Squadron at Tyndall AFB in Panama City, FL. In this role, he oversaw the modification of the QF-4 in support of various operational missile testing efforts. After three years at Tyndall, Clay was assigned as the Launch Verification Engineer for the Evolved Expendable Launch Vehicle Program Office at Los Angeles AFB in California. He managed a 2000 item checklist which ensured each rocket had been appropriately verified and validated and was ready for launch, including the launch of the first ever Delta IV Heavy rocket.

In 2006, he was accepted to the Air Force Institute of Technology at Wright Patterson AFB, in Dayton, Ohio. There, he began a Master of Science in Systems Engineering, specializing in Space Systems and Command, Control, Communication, Computers, Intelligence, Surveillance, Reconnaissance (C4ISR).

Upon graduation from the Air Force Institute of Technology in 2008 he was assigned to the Air Vehicles Structures Division at the Air Force Research Laboratory at Wright Patterson AFB. He was the program lead for development of technologies for autonomous reusable first stage booster systems. Clay was then chosen to be the executive officer for the Vice Commander of the Air Force Research Laboratory.

In 2010, Clay moved to his new assignment at the National Reconnaissance Office (NRO). He began his role there as the program manager for a \$ 500 million engineering support contract, then was the program manager for an airborne LIDAR asset providing support for real world operational requirements.

He moved back to Wright Patterson in 2013 to begin his PhD at AFIT.

REPORT DOCUMENTATION PAGE					Form Approved OMB No. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. <b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>						
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From — To)		
15-10-2016		Doctoral Dissertation		Sept 2013 — Sep 2016		
4. TITLE AND SUBTITLE  Optimal Control of an Uninhabited Loyal Wingman				5a. CONTRACT NUMBER		
				5b. GRANT NUMBER		
				5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)  Humphreys, Clay J., Major, USAF				5d. PROJECT NUMBER		
				5e. TASK NUMBER		
				5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER  AFIT-ENY-DS-16-S-063		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory, Aerospace Systems Directorate (AFRL/RQ) Mrs. Amy Burns 2130 Eighth Street, Building 45, Room 190 WPAFB, OH 45433-7765				10. SPONSOR/MONITOR'S ACRONYM(S)  AFRL/RQQC		
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT  DISTRIBUTION STATEMENT A: APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.						
13. SUPPLEMENTARY NOTES						
14. ABSTRACT This work demonstrates the use of optimal control and stochastic estimation techniques as an autonomous near real-time dynamic route planner for the DoD concept of the loyal wingman. First, the optimal control problem is formulated for a static threat environment and a hybrid numerical method is demonstrated. The optimal control problem is transcribed to a nonlinear program using direct orthogonal collocation, and a heuristic particle swarm optimization algorithm is used to supply an initial guess to the gradient-based nonlinear programming solver. Next, a dynamic and measurement update model and Kalman filter estimating tool is used to solve the loyal wingman optimal control problem in the presence of moving, stochastic threats. Finally, an algorithm is written to determine <i>if</i> and <i>when</i> the loyal wingman should dynamically re-plan the trajectory based on a modified distance to re-plan formulation which uses speed and stochastics of the moving threat as well as relative distance and angle of approach of the loyal wingman to the threat. Results demonstrate a methodology for rapidly computing an optimal solution to the loyal wingman optimal control problem.						
15. SUBJECT TERMS Optimal Control, Loyal Wingman, Unmanned Wingman, Pseudospectral Method, Direct Orthogonal Collocation, Particle Swarm Optimization, Dynamic Re-planning, Hybrid Optimal Control						
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON	
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Richard G. Cobb, AFIT/ENY	
U	U	U	U	159	19b. TELEPHONE NUMBER (include area code) (937) 255-3636, x4559; richard.cobb@afit.edu	